
DIPLOMARBEIT

Herr
Matthias Groß

**Automatisierung einer
3D-Messanlage zur Evaluierung
von Hörgeräten**

Mittweida, 2010

DIPLOMARBEIT

Automatisierung einer 3D-Messanlage zur Evaluierung von Hörgeräten

Autor:

Herr

Matthias Groß

Studiengang:

Mechatronik

Seminargruppe:

ME05w1

Erstprüfer:

Prof. Dr.-Ing. Dietmar Römer

Zweitprüfer:

Dr. Volker Gebhardt

Einreichung:

12. November 2010

Verteidigung/Bewertung:

Mittweida, 2010

Bibliografische Angaben:

Groß, Matthias:

Automatisierung einer 3D-Messanlage zur Evaluierung von Hörgeräten - 2010 - 98, 6, 87 S.

Mittweida, Hochschule Mittweida (FH), University of Applied Sciences,

Fakultät Maschinenbau, Diplomarbeit, 2010

Kurzfassung:

Die vorliegende Arbeit beschäftigt sich mit der Automatisierung einer Messanlage, die die Kommunikation zwischen zwei Hörgeräten in Abhängigkeit von geräteinternen, elektromagnetischen Störungen untersucht. Realisiert wird dies durch die Ermittlung der maximalen Sendereichweite der Geräte, um so Rückschlüsse auf das Maß der Störwirkung zu ziehen. Zweck der Anlage ist es, die kritischen Bauteile eines Hörgerätes bezüglich ihrer Position analysieren zu können. Um dies zu systematisieren, wurden zum Einen 2 Schrittmotoren zur Positionierung auf XY-Ebene, und ein weiterer zur Ermittlung der Sendereichweite auf Z-Ebene integriert. Zielstellung war es, die vorhandene Hard- und Software so zu erweitern, dass ein selbständiger Prozess entsteht, der das Einwirken des Benutzers weitgehend ausschließt. Unter Konsultation der Anlagenbenutzer wurde zunächst ein Konzept aufgestellt, das Anforderungen, Wünsche und Möglichkeiten des Gesamtsystems beleuchtet. Bei der Umsetzung wurde besonders darauf geachtet, die Prozesszeit gering zu halten und gleichzeitig alle auftretenden Situationen vorhersehbar zu machen. Die Programmierung der Software erfolgte in Zusammenarbeit mit dem Hauptanwender. Nach zahlreichen Tests der entwickelten Software mit den erweiterten Hardwarekomponenten konnte ein sicherer Betrieb der Anlage sowie die enorme Ersparnis von Zeit und Aufwand erzielt werden. Parallel dazu wurden Punkte zur Weiterentwicklung erfasst und aufgestellt.

Abstract:

The present study investigates the automation of a measurement system which evaluates the communication between hearing aids depending on internal disturbances. To deduce from the amount of the disturbances the hearing devices are tested on their maximum transmission range. The purpose of the measurement system is to analyse components with interfering behaviour referring to their position. There are two stepper motors for positioning on the XY-level and an additional one for determining the transmission range on the Z-axis. To plan on an autonomous process without user actions, the hardware and software components have been expanded. Consulting the future users, a concept was raised to focus on requirements, wishes and capability of the system. Within the process of assembling, low process time and the predictability of all possible situations has been esteemed. The software programming was performed in cooperation with the main users. After numerous tests of the newly implemented software with the adapted hardware components, a safe operation and savings of time and effort on this system have been established. Additionally, issues of further development have been worked out.

Inhaltsverzeichnis

1	Einleitung	10
2	Grundlagen.....	12
2.1	Binaural gekoppelte Hörsysteme	12
2.2	Grundlagen zur Messung	14
2.2.1	Aufbau	14
2.2.1.1	Faceplates.....	15
2.2.2	Koordinatensysteme	15
2.2.3	Ablauf	16
2.2.3.1	Pingpong-Test.....	19
2.2.3.2	Trimmung der Hörgeräte	19
3	Stand der Technik - Hardware	20
3.1	Übersicht	20
3.2	Grundplatte	22
3.3	Schrittmotorsteuerung.....	22
3.3.1	Aufbau	22
3.3.2	Konfiguration und Programmierung	24
3.3.3	Datenübertragung.....	25
3.3.4	Netzteil	26
3.4	XY-Rahmen	27
3.5	X- und Y-Schieber	27
3.6	X- und Y-Spindel	28
3.7	X- und Y-Schrittmotor.....	29
3.7.1	Aufbau und Funktion.....	29
3.7.2	Umrechnung der Drehzahl	30
3.8	Endlagenschalter	31
3.9	XY-Schiff.....	31

3.10	Z-Rahmen	32
3.11	Z-Spindel.....	32
3.12	Z-Träger	33
3.12.1	Aufbau und Funktion.....	33
3.13	Z-Schrittmotor	33
3.13.1	Aufbau und Funktion.....	33
3.13.2	Umrechnung der eingestellten Drehzahl	34
3.13.3	Umrechnung des Haltemomentes	35
3.14	Referenzgerät	35
3.15	Messsonde	36
3.16	Standalone Wireless Tester.....	38
3.17	Testfaceplate	39
4	Stand der Technik - Software.....	40
5	Anforderungen und aktueller Stand im Vergleich	42
5.1	Hardware.....	42
5.1.1	Konfiguration.....	42
5.1.2	Feineinstellung der Messspule.....	42
5.1.3	Z-Achse	42
5.2	Messungen	43
5.2.1	Definition der Messpunkte.....	43
5.2.2	Pingpong-Test.....	43
5.2.3	Automatische Trimmung.....	43
5.2.4	Messablauf	44
5.2.5	Datenspeicherung und -auswertung.....	44
5.3	PC-Benutzeroberfläche	44
5.3.1	Bedienung.....	44
5.3.2	Fahrfunktionen.....	45
5.3.3	Statusüberwachung.....	45

5.3.4	Einstellmöglichkeiten	45
6	Umsetzung der Anforderungen	46
6.1	Hardware	46
6.1.1	Stufenlose Höheneinstellung der Messspule	46
6.1.2	Einführung des Wireless Programmer	47
6.1.3	Netzteile	50
6.1.4	Z-Schrittmotor	51
6.1.4.1	Dimensionierung	51
6.1.4.2	Einstellung der Drehzahl	54
6.1.5	Webcam	55
6.2	Software	56
6.2.1	Kommunikation mit der Motorsteuerung	58
6.2.1.1	Schnittstellendefinition	58
6.2.1.2	Senden von Maschinenbefehlen	58
6.2.2	Realisierung einfacher Fahrbefehle	60
6.2.2.1	Freies Verfahren der Achsen	60
6.2.2.2	Koordinierte Bewegung der Achsen	61
6.2.3	Erstellung von Fahrzyklen	61
6.2.3.1	Vorgabe von XY-Pfaden	61
6.2.3.2	Konstruktion einer Abfolge für die XYZ-Positionierung	62
6.2.3.3	Erzeugung einer Kalibrieroutine	63
6.2.4	Kommunikation zwischen PC und Hörgerät	64
6.2.5	Automatisches Trimmen	64
6.2.6	Kombination von Pingpong-Test und Positionierung	65
6.2.7	Datenverarbeitung und Speicherung	68
6.2.7.1	Erstellung einer normierten Messdatenstruktur	68
6.2.7.2	Aufbau einer Konfigurationsstruktur	69
6.2.8	Komplettierung, Einbeziehen des Benutzers	70
6.2.8.1	Grafische Definition des Messbereichs	70
6.2.8.2	Erstellung der Benutzeroberfläche	75
7	Umsetzungen und Anforderungen im Vergleich	82
7.1	Hardware	82

7.1.1	Konfiguration.....	82
7.1.2	Feineinstellung der Messspule.....	82
7.1.3	Z-Achse	82
7.2	Messungen	82
7.2.1	Definition der Messpunkte.....	82
7.2.1.1	Pingpong-Test.....	82
7.2.2	Trimmen der Hörgeräte	83
7.2.3	Messablauf	83
7.2.4	Datenspeicherung und –auswertung.....	83
7.3	PC-Benutzeroberfläche	83
7.3.1	Bedienung.....	83
7.3.2	Fahrfunktionen.....	84
7.3.3	Statusüberwachung.....	84
7.3.4	Einstellmöglichkeiten	84
8	Ausblick und Erweiterungen	85
	Anhang	87
A	Datenblätter.....	87
A.1	Nanotec L4018S161.....	88
A.2	Trinamic QSH5718-51-28-101 (Auszug aus [15])	89
A.3	Trinamic TMCM-310 (Auszug aus [4])	91
B	TMCL-Befehlsliste (Auszug aus [5])	92
C	MATLAB-Funktionsliste	93
D	Datenträger.....	97
	Abkürzungsverzeichnis.....	98
	Literaturverzeichnis.....	99
	Erklärung.....	101
	Danksagung	102

1 Einleitung

Die Funktionalität von Hörgeräten umfasst in der heutigen Zeit weitaus mehr als das bloße Verstärken von Umgebungsgeräuschen. Mit dem Einzug digitaler Technologien konnte deren Funktionsumfang um ein Vielfaches erweitert werden. Neben der Abstimmung auf verschiedene akustische Umgebungsszenarien können auch Signale verschiedener Audioquellen (Mobiltelefon, Mp3-Player etc.) über eine induktive Funkstrecke empfangen werden.

Gleichzeitig zur Entwicklung auf funktioneller Seite steigt auch die Miniaturisierung der Geräte, da die Zielsetzung, das Gerät möglichst „unsichtbar“ zu machen, von essentieller Bedeutung ist.

Einen weiteren wichtigen Fortschritt in der Hörgeräteentwicklung bildet die kabellose Übermittlung von Steuersignalen zwischen rechtem und linkem Hörgerät (Wireless-e2e; e2e = ear-to-ear). Realisiert wird dies durch Funkübertragung im unteren MHz-Bereich. Hörsysteme, die mit dieser Technik ausgestattet sind, sogenannte binaural gekoppelte Hörgeräte, können damit Informationen bezüglich der Hörumgebung austauschen und sich entsprechend synchronisieren [1].

Resultierend aus dem Prinzip der Wireless-Technologie und der steigenden Packungsdichte der Bauteile kann es geräteintern jedoch zur Entwicklung von Störstrahlungen kommen, die die Sende- und Empfangsfähigkeit stark beeinträchtigen [2].

Um dies schon vor der Fertigung der Geräte begutachten zu können, existiert in der Entwicklungsabteilung des Siemens Audiologischen Technik eine Anlage, die die Störwirkung einzelner Bauteile oder Baugruppen untersucht. Ziel hierbei ist es, eine Charakterisierung der direkten Umgebung des Bauteils aufzustellen, um Aussagen hinsichtlich Bauteilanordnung und Maßnahmen zur Störungsvermeidung treffen zu können.

Geht man von der Forderung aus, die Umgebung eines Bauteils in Millimeterabständen dreidimensional analysieren zu können, ergibt sich daraus ein beträchtlicher Umfang an einzelnen Messpunkten. Mit der vorhandenen Anlage ist der Benutzer bereits in der Lage jeden Messpunkt einzeln und manuell untersuchen zu können, was jedoch einen immensen Zeit- und Bedienaufwand zur Folge hat.

Ziel dieser Diplomarbeit ist es nun, die Anlage zu automatisieren, damit die ständige Interaktion durch den Benutzer nicht mehr erforderlich ist.

In Kapitel 2 werden zunächst die Grundlagen, auf denen das Anlagenkonzept basiert, erläutert. Dabei wird der Fokus auf die Gerätschaft und die zugrunde liegende Messmethodik gelegt.

Kapitel 3 und 4 stellen den Stand der zu Beginn der Diplomarbeit vorhandenen Hardware- und Softwaretechnik dar. Darin werden die Komponenten der Anlage näher beschrieben, um die später durchgeführten Veränderungen zu verdeutlichen.

Kapitel 5 umfasst eine Auflistung von Anforderungen an die Automatisierung im Vergleich zum aktuellen Stand. Darin sollen dem Leser die notwendigen Verbesserungen dargelegt werden.

In Kapitel 6 wird die Umsetzung dieser Anforderungen veranschaulicht. Dabei werden alle hinzugefügten bzw. ersetzten Hardwarekomponenten gezeigt, sowie die wichtigsten Funktionen der neuen Software erläutert. Im Softwareteil dieses Kapitels werden einige der in MATLAB programmierten Funktionen genannt. Weiterhin werden dort Befehlskürzel aus der Programmiersprache der Motorsteuerung (TMCL, s [5]) aufgeführt, die durch *kursive GROSSBUCHSTABEN* hervorgehoben werden.

Kapitel 7 zeigt noch einmal die umgesetzten Punkte im Vergleich zu den Anforderungen aus Kapitel 5.

Das letzte Kapitel gibt einen Ausblick auf Erweiterungen, die zur weiteren Steigerung der Anlageneffizienz beitragen könnten.

Anhang A enthält die Datenblätter der elektrischen Geräte. In Anhang B sind alle Steuerbefehle der TMCL-Programmiersprache gelistet. Anhang C beinhaltet die in MATLAB erstellten Funktionen.

2 Grundlagen

2.1 Binaural gekoppelte Hörsysteme

Wie bereits einleitend erwähnt, kommunizieren binaural gekoppelte Hörgeräte untereinander über Funk. Des Weiteren wird die Wireless-Technologie zur Kommunikation zwischen Hörgerät und dessen Fernbedienung eingesetzt. Systeme dieser Art ermöglichen es u.A.

- den Geräten, sich untereinander eigenständig zu synchronisieren,
- an einem Hörgerät durchgeführte Benutzereinstellungen (wie Lautstärkeänderung, Programmumschaltung) an das zweite Gerät zu übertragen,
- Audioquellen an die Fernbedienung anzuschließen und die Signale an die Hörgeräte zu übertragen.

Hörgeräte und Fernbedienung verfügen dabei über eine eingebaute Ferritantenne („Wireless-Spule“) die als Sende- und Empfangseinheit dient. Die Datenübertragung wird über Phasenmodulation des magnetischen Nahfeldes realisiert [2].

Durch die hohe Bauteilintegration in den Hörgeräten können jedoch Störeinflüsse, die durch benachbarte Komponenten hervorgerufen werden, die Sende- und Empfangseigenschaften negativ beeinflussen. Man unterscheidet dabei zwischen aktiven und passiven Störungen. Die folgende Übersicht stellt die kritischen Komponenten und deren Störungsart der Beeinflussung dar.

Batterien und Metallteile

Metallische Komponenten sowie Batterien wirken als *passive Störer*. Sie verfügen dabei über kein Störfeld sondern „verbiegen“ und dämpfen das Magnetfeld der Wireless-Spule durch ihre elektrischen und magnetischen Eigenschaften. Diese Problematik wirkt sich hauptsächlich auf Güte und Resonanzfrequenz des Empfangskreises aus.

Hörer

Der Lautsprecher eines Hörgerätes wird als Hörer bezeichnet. Durch das Magnetfeld, das die darin eingebaute Spule abstrahlt, wirkt er als *aktiver Störer*. Die störende Induktivität wird hierbei durch die Einkopplung des Magnetfeldes der Hörerspule in das Feld der Wireless-Spule hervorgerufen.

Hybride

Der Hybrid ist die wichtigste Elektronikkomponente im Hörgerät. Er umfasst eine kompakt angeordnete Kombination aus analogem und digitalem Chip. Hervorgerufen durch die Taktfrequenzen der Signalverarbeitung sowie die Beschaltung der Chips entwickelt er ein *aktives Störfeld*, das wiederum in die Wireless-Spule eingekoppelt wird und dadurch das Funksystem stört.

Platine und Leiterschleifen

Bei größeren Hörgeräten (hauptsächlich HdO) werden statt den kompakten Hybriden Platinen verwendet, die analog dazu *aktive Störer* sind. Des Weiteren können auch die auf den Platinen aufgetragenen Leiterbahnen Störfelder erzeugen, wenn sie untereinander Schleifen bilden.

Abbildung 1 zeigt zwei verschiedene Hörgerätetypen mit den darin verbauten Wireless-Spulen.

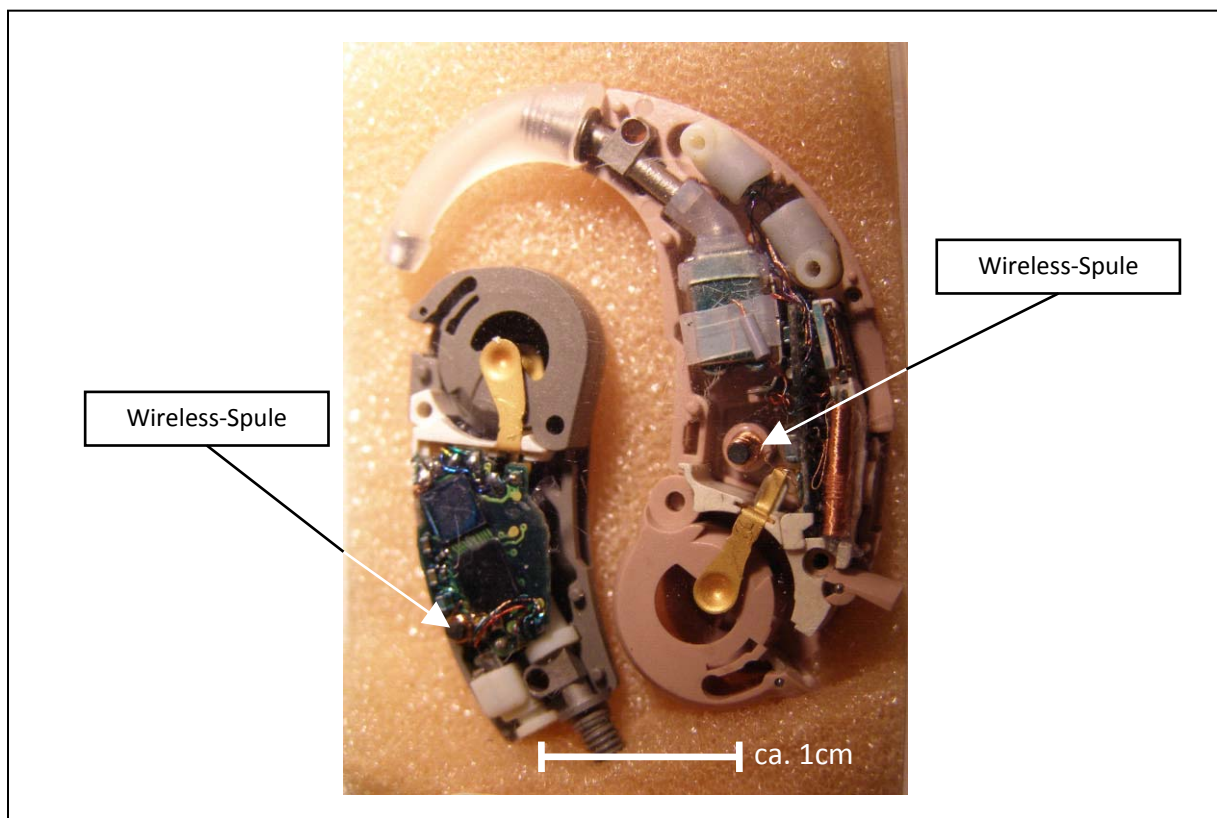


Abbildung 1: Wireless-Spule an zwei Hörgeräten [3]

Abbildung 2 zeigt die Computersimulation der Magnetfelder einer Batterie und eines Hörers und deren Einkopplung in die Wireless-Spule.

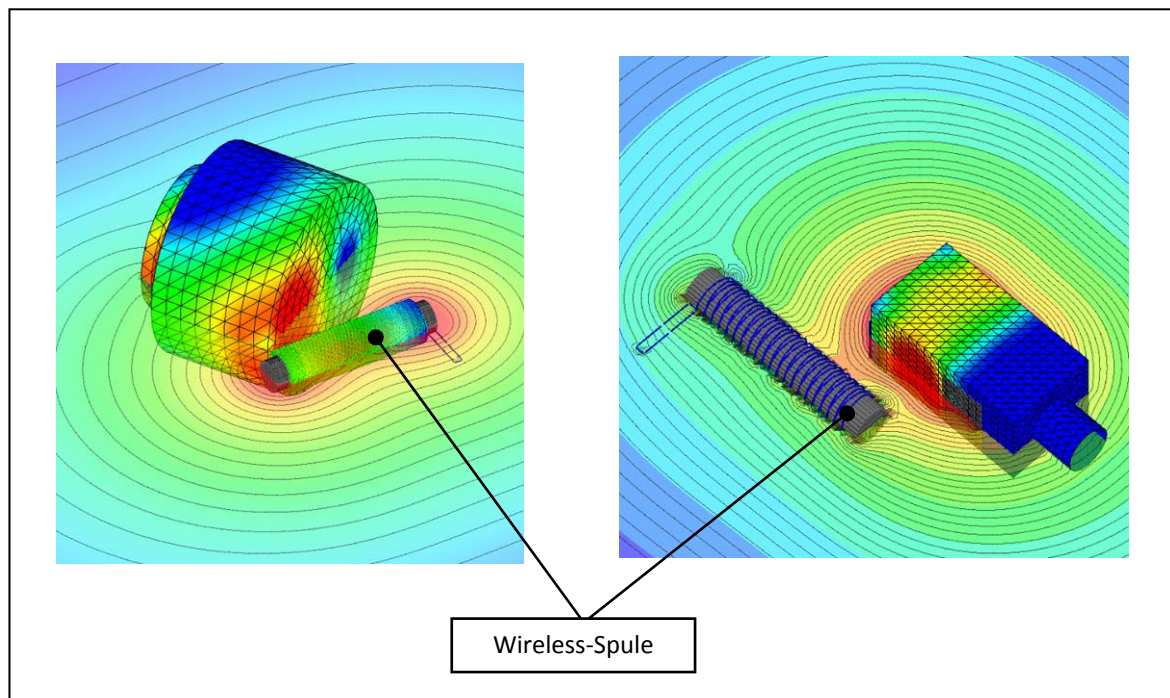


Abbildung 2: FEKO-Feldsimulation Wireless-Antenne nahe einer Batterie (l.) und eines Hörers (r.) [2]

Um die Wirkung dieser Störfelder möglichst gering zu halten, müssen die einzelnen Komponenten in ihrer dreidimensionalen Umgebung ausreichend auf ihre Störwirkung untersucht werden. Entscheidend ist dabei auch die relative Anordnung der Komponente zur Antenne. Für eine aussagekräftige Messmethodik muss daher genügend Flexibilität hinsichtlich der Messpositionen gewährleistet sein.

2.2 Grundlagen zur Messung

2.2.1 Aufbau

Bei den Messungen soll ermittelt werden, welche Bauteile an welchen Positionen die Kommunikation zwischen Sender und Empfänger durch aktive oder passive Störungen behindern.

Der grundlegende Messaufbau setzt sich jeweils zusammen aus

- den zu untersuchenden Bauteilen oder –gruppen (Testobjekte mit zugehöriger Funktionselektronik)
- der Messsonde (bestehend aus Messspule und Elektronik), die der Antenneneinheit eines Hörgerätes entspricht
- dem Referenzgerät, welches als auswertendes Hörgerät dient und
- einer Benutzerfernbedienung zur Vorabstimmung der Hörgeräte

Testobjekt und Referenzgerät sowie die Elektronik der Messsonde liegen grundsätzlich ohne Gehäuse auf sog. Faceplates vor (s. 2.2.1.1). Bezüglich der Positionen muss zwischen zwei Koordinatensystemen unterschieden werden (s. 2.2.2)

2.2.1.1 Faceplates

Ein Faceplate ist ein spezieller Kunststoffträger für die IdO-Fertigung, auf dem elektronische Komponenten aufgebracht werden. Ein komplettes IdO besteht aus dem Faceplate und einer Kunststoffschale, die dem Ohrabdruck des Patienten entspricht. Für die Anlage sind die einheitliche Größe und der einfache Zugang zu den Bauteilen auf Faceplates von Vorteil.

Für die gängigsten Bauteile gibt es zusätzliche Kunststoffträger, auf denen das Bauteil in größerem Abstand zu seiner Stromversorgung neben dem Faceplate fixiert werden kann.

2.2.2 Koordinatensysteme

Unter Betrachtung der Teilschritte des Messablaufes (s. 2.2.3) muss hinsichtlich der Positionierungen zwischen zwei Koordinatensystemen unterschieden werden:

- dem Anlagenkoordinatensystem und
- dem Objektkoordinatensystem.

Die Positionierung auf der XY-Ebene ist bei beiden Koordinatensystemen identisch, d.h. eine Bewegung auf der X- oder Y-Achse der Anlage entspricht in gleichem Maße der X- oder Y-Bewegung am Testobjekt.

Bezüglich der Z-Achse muss jedoch unterschieden werden:

Die Positionierung auf der *Anlagen-Z-Achse* betrifft den Abstand zwischen Referenzgerät und Messsonde, der zur Ermittlung der maximalen Sendereichweite verändert wird. Im weiteren Verlauf dieser Arbeit wird die Anlagen-Z-Achse als Z-Achse bezeichnet.

Die *Objekt-Z-Achse* umfasst die eingestellte Höhe der Messspule am Testobjekt („Eintauchtiefe“). Da die Umgebung des Testobjektes dreidimensional untersucht werden soll, ist zusätzlich zur Positionierung auf der XY-Ebene eine Veränderung der Messspulenposition in Z-Richtung erforderlich.

2.2.3 Ablauf

Der Messablauf besteht im Allgemeinen aus den folgenden Teilschritten:

- der Positionierung der Messsonde auf dem XY-Millimeterraster
- der Positionierung des Referenzgerätes auf der Z-Achse in Zentimeterschritten
- einem Verbindungstest zwischen Referenzgerät und Messsonde, dem sogenannten Pingpong-Test und
- der optionalen Höhenverstellung der Messspule auf der Objekt-Z-Achse.

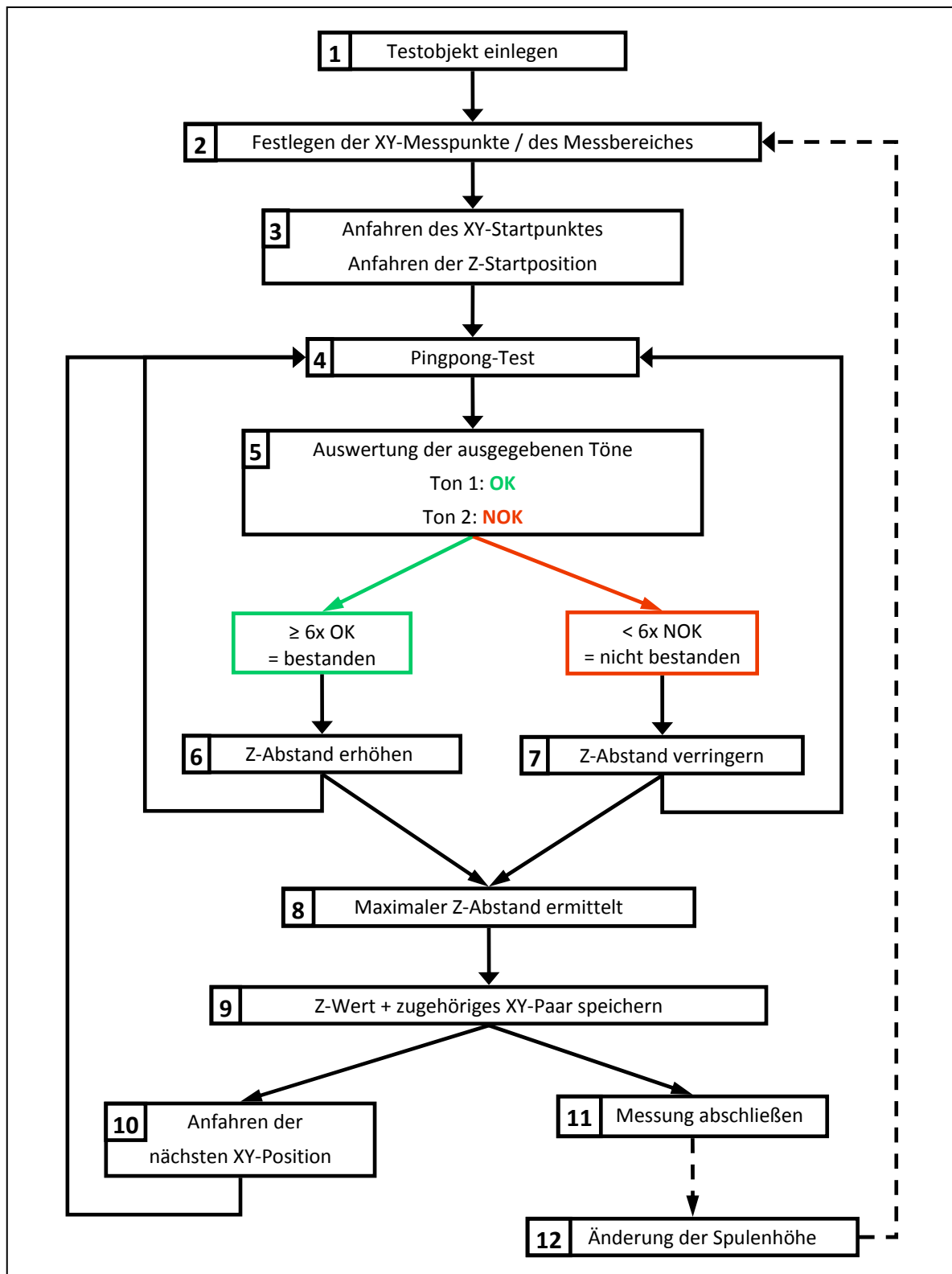


Abbildung 3: Messablaufplan

Schritt 1

Vor jeder Messung wird das Testobjekt auf dem Faceplate im unteren Teil der Anlage eingelegt.

Schritt 2

Grundlage jeder Messung ist eine, vom Nutzer vordefinierte Messfläche auf der XY-Ebene. Aus dieser Fläche ergibt sich ein Millimeterraster, welches die XY-Punkte definiert, die die Messsonde anfährt.

Schritt 3

Die Messsonde wird am ersten Messpunkt positioniert. Analog dazu wird das Referenzgerät auf der Z-Achse an eine Startposition gefahren.

Schritt 4

An jeder XY-Position wird die Kommunikation zwischen Referenzgerät und Messsonde mit dem Pingpong-Test (s. 2.2.3.1) überprüft.

Schritt 5

Für die Auswertung des Pingpong-Tests gilt generell: Nach 6 erfolgreichen Pingpong-Vorgängen (6 mal Ton „OK“) gilt der Test als bestanden.

Schritt 6, Schritt 7

Je nach Testergebnis wird durch zentimeterweises Verfahren des Referenzgerätes auf der Z-Achse dessen Abstand zur Messsonde erhöht oder verringert. Anschließend wird der Test wiederholt.

Schritt 8

Der Pingpong-Test wird so oft wiederholt bis die maximale Sendereichweite ermittelt wurde, d.h. der maximale Abstand, an dem der Pingpong-Test mindestens 6 mal als erfolgreich gewertet wurde.

Schritt 9

Der Z-Wert wird zusammen mit der X- und Y-Position gespeichert. Durch die dabei entstehende 3D-Matrix kann der Benutzer nach Abschluss der Messung die Störwirkung des Testobjektes flächenmäßig beurteilen.

Schritt 10

Nach dem Datenspeichern wird die nächste XY-Position angefahren und der Test wiederholt.

Schritt 11

Sobald der Test an allen XY-Punkten durchgeführt wurde und die Daten entsprechend übernommen wurden, können diese exportiert und ausgewertet werden.

Das Messergebnis besteht aus einer 3D-Matrix, die die maximale Sendereichweite an den verschiedenen Positionen der XY-Umgebung des Testobjektes angibt.

Schritt 12

Nach Abschluss einer Messung kann die Position der Messspule auf der Objekt-Z-Achse verändert werden. Dies dient dem Zweck, die XY-Messfläche auf eine andere Z-Position zu verschieben oder dort auch neu zu definieren.

2.2.3.1 Pingpong-Test

Der Verbindungstest zwischen Referenzgerät und Messsonde wird als Pingpong-Test bezeichnet. Das Referenzgerät sendet dabei periodisch Signale an die Messsonde („Ping“). Falls dieses sich in Reichweite befindet, sendet es ein Antwortsignal zum Referenzgerät zurück („Pong“). Gehen von den zu untersuchenden Bauteilen zu große Störungen aus, wird das Empfangsvermögen der Messsonde so weit reduziert, dass diese außer Reichweite des Referenzgerätes gerät und kein Antwortsignal senden kann.

Durch die Ausgabe zweier unterschiedlicher Töne bestätigt das Referenzgerät den Empfang oder Nicht-Empfang eines Antwortsignals nach jedem Sendevorgang. Das Intervall zwischen zwei Tönen beträgt ca. 1 Sekunde.

2.2.3.2 Trimmung der Hörgeräte

Das Einwirken passiver Störungen (s. 2.1) kann eine Verstimmung der Übertragungsfrequenz der Hörgeräte verursachen. Um diesen Fehler zu beheben werden die Geräte getrimmt. Das Trimmen umfasst die Messung und Einstellung von drei, im Funksystem verwendeten Frequenzen (untere Grenzfrequenz, obere Grenzfrequenz und Arbeitsfrequenz) und wird mit dem Programmiergerät (SWT, s. 3.16) durchgeführt. Dieser Vorgang dauert ca. 4 Sekunden und muss während der Messung regelmäßig durchgeführt werden.

3 Stand der Technik - Hardware

3.1 Übersicht

Zu Beginn dieser Diplomarbeit stand laborseitig nur eine Anlage für Messungen bereit. Um den Durchsatz an Messungen zu erhöhen, wurde eine zweite, baugleiche Anlage gebaut, die sich im Wesentlichen nur durch Länge und Durchmesser der Z-Gewindespindel von der ersten Anlage unterscheidet (s. 3.11). Alle Träger- und Rahmenteile wurden in der betriebsinternen Werkstatt konstruiert und gefertigt. Die elektrischen Komponenten sind extern bestellte Zukaufteile.

Zur Vorbeugung elektromagnetischer Störfelder und Verkoppelungen wurde bei der Entwicklung darauf geachtet, dass möglichst nur Kunststoffteile verwendet werden. Unabdingbare Metallkomponenten wie Motoren und Verkabelung wurden daher auf der größer dimensionierten Grundplatte weitgehend vom Messaufbau distanziert.

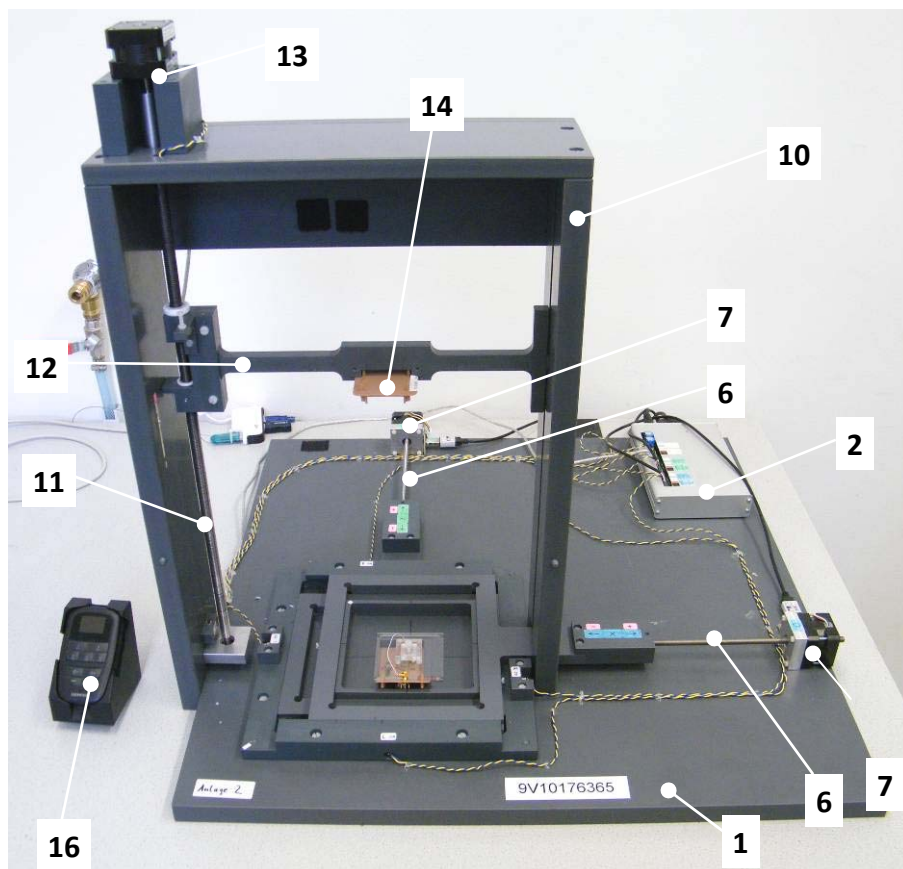


Abbildung 4a: Gesamtansicht

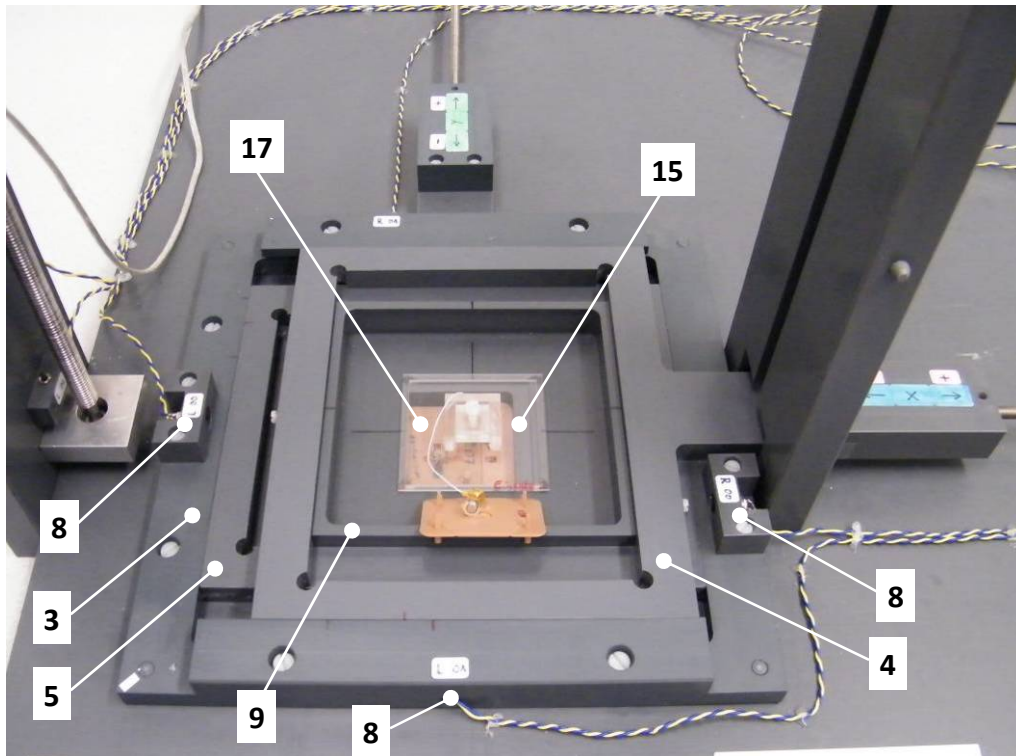


Abbildung 4b: Messaufbau

Tabelle 1: Übersicht über die Anlagenkomponenten

Nr.	Komponente	Kurzbeschreibung
1	Grundplatte	PVC, Maße (mm): 600x600x20
2	Schrittmotorsteuerung	TRINAMIC Steuerkarte mit Gehäuse, Netzteil und Verkabelung
-	Mess-PC	Standard Desktop-PC mit Windows Betriebssystem
X-, Y-Achse		
3	XY-Rahmen	PVC, Maße (mm): 255x255x24
4	X-Schieber	PVC, Maße (mm): 250x250x12
5	Y-Schieber	PVC, Maße (mm): 250x250x12
6	X-, Y-Spindel	Gewinde: M6x0,5; Länge: 200 mm
7	X-, Y-Schrittmotoren	NANOTEC Linearaktuatoren
8	Endlagenschalter	2 Taster pro Achse
9	XY-Schiff	PVC, Maße (mm): 70x70x15
Z-Achse		
10	Z-Rahmen	PVC, dreiteilig, Gesamthöhe (mit Schrittmotor) ca. 61,5 cm

11	Z-Spindel	Anlage 1: M8x1,3, Länge ca. 42 cm, Anlage 2: M10x1,3, Länge ca. 48,5 cm
12	Z-Träger	PVC
13	Z-Schrittmotor	NANOTEC Linearaktuator (wie X- und Y-Achse)
Messaufbau		
14	Referenzgerät	Hörgerät auf Faceplate mit Sendespule
15	Messsonde mit Vorrichtung und Elektronik	Messspule mit Feinjustierung, Koaxialkabelverbindung zum Faceplate
16	Standalone Wireless Tester	Benutzerfernbedienung für die Trimmung der beiden Hörgeräte
17	Testfaceplate	Faceplate mit Testobjekten

Die Positionierung erfolgt an der bisher vorhandenen Anlage an allen drei Achsen manuell. Über ein grafisches Userinterface am PC kann der Motor in Millimeterschritten (X- und Y-Achse) oder Zentimeterschritten (Z-Achse) verfahren werden.

Der Pingpong-Test muss vom Benutzer akustisch überwacht werden. Das Referenzgerät gibt dabei im Sekundentakt Audiosignale aus, die sich ändern, sobald sich ein zweites, „antwortendes“ Gerät in Reichweite befindet.

3.2 Grundplatte

Die Maße der Grundplatte wurden gezielt höher gewählt, um die Motoren und die Schrittmotorsteuerung weitgehend vom Messaufbau isolieren zu können. Dies dient dazu, den Einfluss der elektromagnetischen Störungen dieser Komponenten von Grund auf zu minimieren.

3.3 Schrittmotorsteuerung

3.3.1 Aufbau

Das TRINAMIC TMCM-310 Steuermodul ist eine programmierbare Steuerkarte für drei zweiphasige Schrittmotoren. Neben einem RS232-Anschluss verfügt die Platine über je 8 In- und Outputs die frei belegt werden können [4]. Die Motoren sowie deren Endlagenschalter sind über 4-polige Steckverbinder an der Oberseite der Karte angeschlossen. Die im Gerät eingestellte Schrittauflösung beträgt 16 Mikroschritte pro Vollschritt (Maximalwert).

Zum Schutz ist die Karte in einem betriebsintern gefertigten Gehäuse untergebracht und wurde, wie die Motoren, in möglichst großem Abstand zum Messaufbau auf der Grundplatte positioniert.

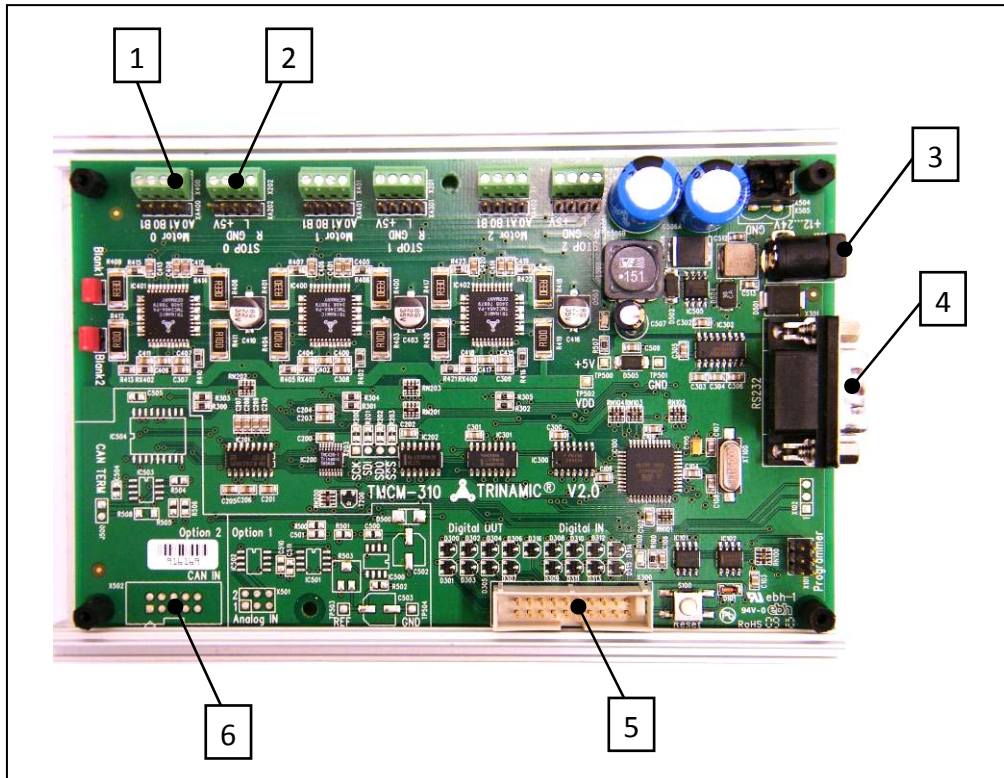




Abbildung 5: TRINAMIC TCM-310

Tabelle 2: TMCM-310 Standardanschlüsse

Nr.	Bezeichnung	Details												
1	Motoranschluss	 <table border="1"> <tr> <td>A0, A1</td><td>-</td><td>Spule 1</td></tr> <tr> <td>B0, B1</td><td>-</td><td>Spule 2</td></tr> </table>	A0, A1	-	Spule 1	B0, B1	-	Spule 2						
A0, A1	-	Spule 1												
B0, B1	-	Spule 2												
2	Anschluss für Endlagenschalter	 <table border="1"> <tr> <td>R</td><td>-</td><td>Rechts</td></tr> <tr> <td>L</td><td>-</td><td>Links</td></tr> <tr> <td>GND</td><td>-</td><td>Masse</td></tr> <tr> <td>+5V</td><td>-</td><td>5V Spannungsversorgung (nicht belegt)</td></tr> </table>	R	-	Rechts	L	-	Links	GND	-	Masse	+5V	-	5V Spannungsversorgung (nicht belegt)
R	-	Rechts												
L	-	Links												
GND	-	Masse												
+5V	-	5V Spannungsversorgung (nicht belegt)												
3	Spannungsversorgung	Buchse für Hohlstecker mit 2,1mm Innen- und 5,5 mm Außendurchmesser												
4	RS232-Buchse	Standard D-Sub Buchse, 9-polig												
5	Inputs / Outputs	8 Inputs (digital oder analog belegbar), 8 Outputs (digital), nicht verwendet												
6	CAN-Schnittstelle	optional, für 10-poligen Steckverbinder, nicht verwendet												

3.3.2 Konfiguration und Programmierung

Die Konfiguration und Programmierung erfolgt über ein mitgeliefertes Software-Tool, der TMCL-IDE (Trinamic Motion Control Language – Integrated Development Environment). Anhand vordefinierter Befehle können damit individuelle Fahrprogramme erzeugt werden. Diese können im EEPROM des Gerätes gespeichert werden und ermöglichen somit einen autarken Betrieb [5].

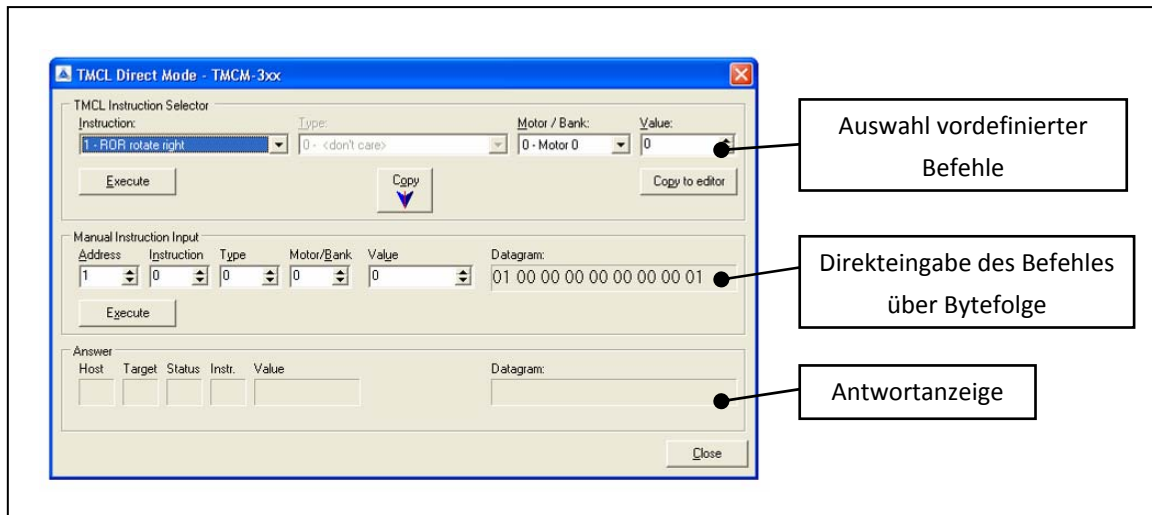


Abbildung 6: Konfigurationstool des TMCM

3.3.3 Datenübertragung

Alle in der Software enthaltenen Befehle können auch, codiert als 9 Byte großer Vektor, direkt über die RS232-Schnittstelle zum Modul gesendet werden. Die Bytefolge des Sendebefehls setzt sich dabei wie folgt zusammen:

Tabelle 3: Zusammensetzung der Sendekommandos [5]

Byte	Beschreibung
1	Moduladresse (generell = 01)
2	Nummer des Befehls
3	Unterkategorie des Befehls
4	Motornummer (0, 1 oder 2)
5	Datenwert zum Setzen numerischer Einstellwerte
6	
7	
8	
9	Prüfsumme

Analog besteht die Antwort des Moduls aus den folgenden Bytes:

Tabelle 4: Zusammensetzung der Byteantwort [5]

Byte	Beschreibung
1	Antwortadresse (generell = 01)
2	Moduladresse (generell = 01)
3	Statusantwort zur Überprüfung des gesendeten Befehls
4	Nummer des Befehls
5	Rückgabewert (z.B. bei Positionsabfrage)
6	
7	
8	
9	Prüfsumme

3.3.4 Netzteil

Das zur Spannungsversorgung der Steuerkarte verwendete Netzteil hat folgende Ausgangsdaten:

Tabelle 5: Netzteilparameter VOLTcraft PA-300 [6]

Parameter	Wert
Ausgangsspannung U_A	12 V
Ausgangsstrom I_A	300 mA
Ausgangsleistung	3.6 W

Nach der Stromteilerregel ergibt sich, pro Motor mit dem Ausgangsstrom I_A des Netzteils und der Anzahl der Phasen N bei Betrieb von nur einem Motor, ein maximal erreichbarer Phasenstrom I_{PH} von:

$$I_{PH} = \frac{I_A}{N} \quad N \text{ Phasenanzahl}$$

$$= \frac{0,3A}{2} = 0,15A$$

Der errechnete Wert des Phasenstromes liegt somit unter dem maximalen Phasenstrom des Motors (0,5 A [7]).

3.4 XY-Rahmen

Der einteilige XY-Rahmen ist mit 4 Kunststoffschrauben an der Grundplatte befestigt. Er dient als Führung der Schieber für X- und Y-Achse und grenzt mit deren Endlagenschalter den Fahrbereich ein.

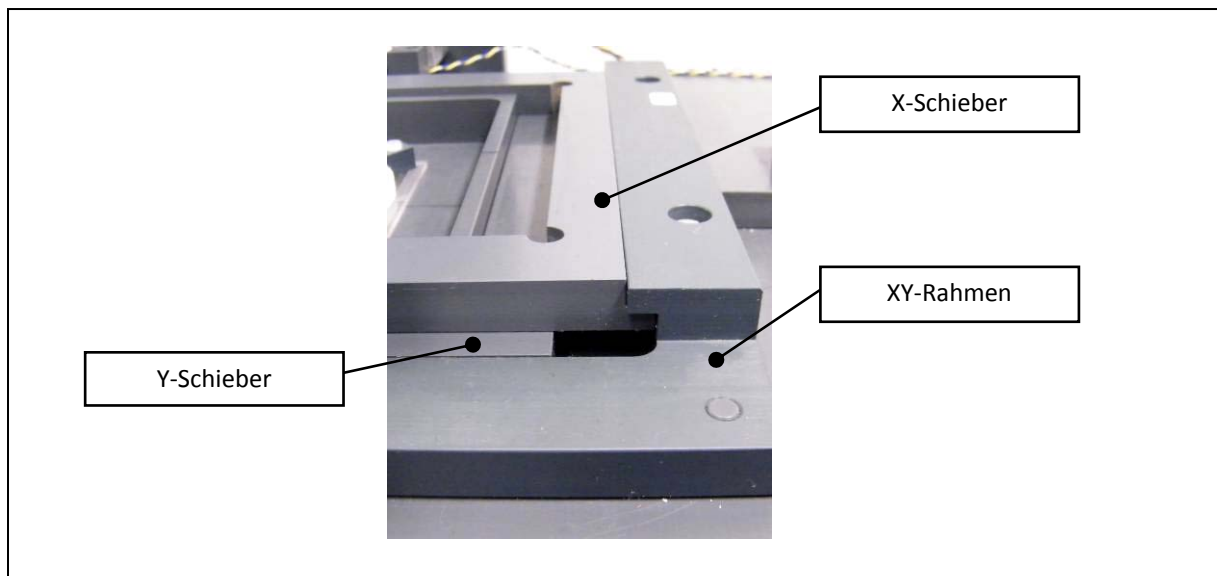


Abbildung 7: Führungskante des XY-Rahmens

3.5 X- und Y-Schieber

Die X- und Y-Schieber führen die Linearbewegung auf der X- und Y-Achse aus. Bestandteil der Schieber ist jeweils ein Rahmenelement und ein damit verschraubter Block, in welchem die Gewindespindel der entsprechenden Achse verdrehsicher fixiert ist (s. Abbildung 8).

Der X-Schieber liegt über dem Y-Schieber. Die Innenfläche, die Beide eingrenzen, entspricht immer der Grundfläche des XY-Schiffes (s. Abbildung 9). Zur Positionierung wird somit das XY-Schiff, in dessen Mittelpunkt die Messspule liegt, an die richtige Position „geschoben“. Nachteil dieses Prinzips ist ein hoher Reibwert zwischen den Schiebern bzw. zwischen Y-Schieber und Grundplatte. Durch die generell kurzen Fahrten und die geringe Belastung kann dies jedoch vernachlässigt werden.

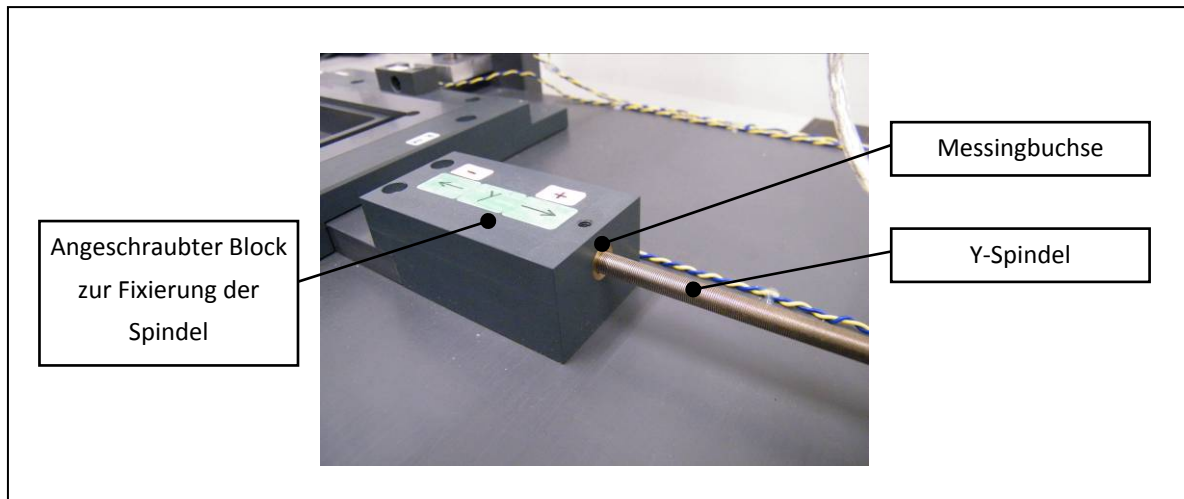


Abbildung 8: Messingbuchse zur Lagerung der Spindel (Y-Achse)

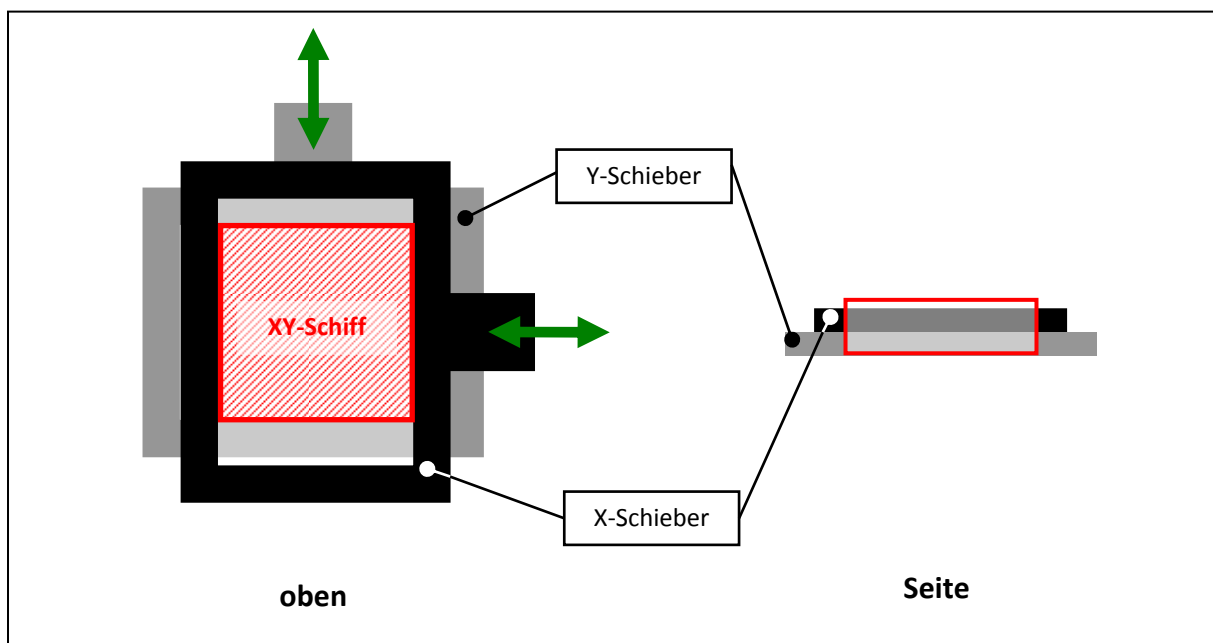


Abbildung 9: Schematische Darstellung der XY-Bewegung

3.6 X- und Y-Spindel

Die Gesamtlänge der Feingewindespindeln für X- und Y-Achse beträgt jeweils 200 mm. Ihre Steigung beträgt 0,5 mm, d.h. mit einer Motorumdrehung legt die Achse einen Axialweg von 0,5 mm zurück.

3.7 X- und Y-Schrittmotor

3.7.1 Aufbau und Funktion

Die Schrittantriebe vom Typ NANOTEC L4018-S1618 sind zweiphasige Hybridschrittmotoren mit einer maximalen Schubkraft von 50 N. Der Rotor verfügt über eine integrierte Gewindebuchse mit einem M6-Feingewinde [8].

Die maximale Auflösung der Motoren beträgt 0,0025 mm pro Vollschrift (entspricht 0,18°). Zusammen mit der maximalen Schrittauflösung der Steuerkarte (16 Mikroschritte pro Vollschrift, [4]) ergibt sich damit eine Gesamtauflösung von 0,156 µm pro Schritt.

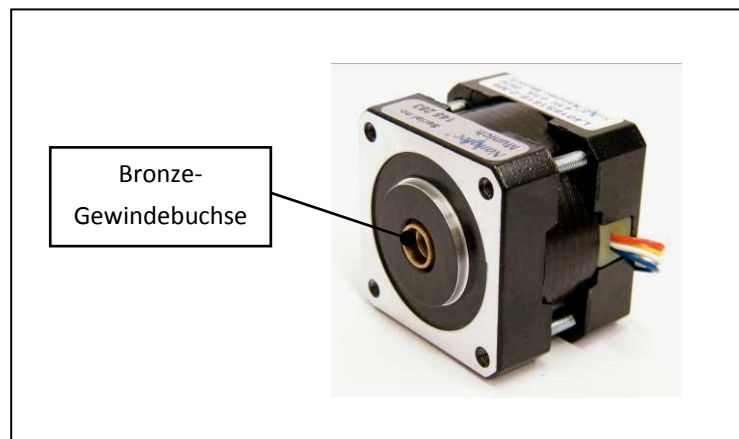


Abbildung 10: NANOTEC L4018-S1618

Abbildung 11 zeigt die Anschlussbelegung an Motor und Steuerkarte.

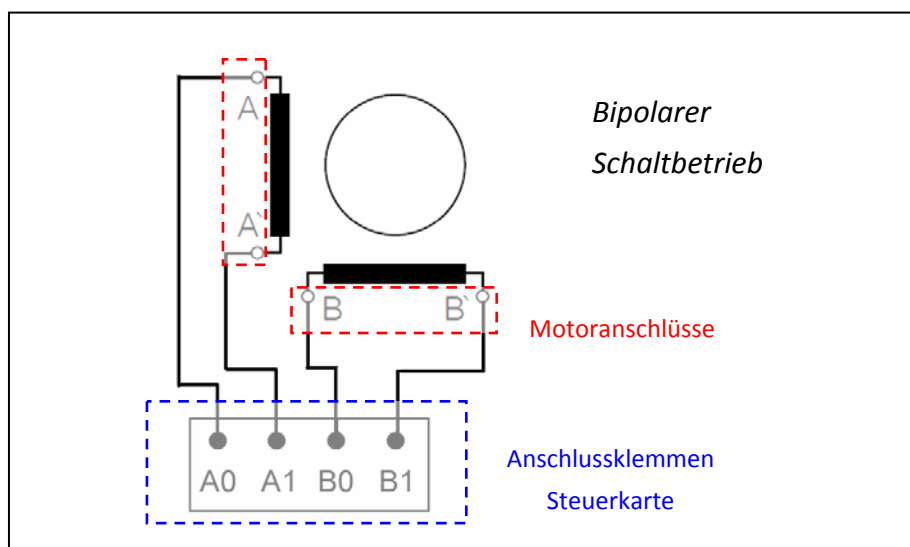


Abbildung 11: Anschlussschema Motor und Steuerkarte [7]

Die Linearbewegung wird durch „Führung“ der Spindel realisiert. Wie bereits in 3.5 beschrieben, ist die Spindel fest am jeweiligen Schieber fixiert. Durch die Rotation der im Motor integrierten Gewindebuchse wird die Spindel damit linear „durch den Motor durch“ bewegt.

3.7.2 Umrechnung der Drehzahl

Die Berechnung der Drehzahl erfolgt mit den im TCM-Steuermodul eingestellten Parametern:

Tabelle 6: Parameter zur Berechnung der X- und Y-Drehzahl

Parameter	Formelzeichen	Wert
TCM-Parameter		
Taktfrequenz	f_{CLK}	16 MHz
Geschwindigkeit	v	250 (gewählt)
Pulse Divisor	pd	3 (gewählt)
Mikroschrittauflösung	S_{MS}	16 Mikroschritte/Vollschritt (gewählt)
Motorparameter		
Schrittauflösung	S_{MOTOR}	200 Vollschritte/Umdrehung

Die TCM-Parameter sind hierbei nur skalierte Zahlenwerte. Um den Wert der reellen Drehzahl zu berechnen, muss der Zahlenwert mit einer speziellen Umrechnungsformel aus [4] berechnet werden.

Zunächst wird aus den Zahlenwerten die Mikroschritt-Frequenz f_{MS} berechnet:

$$\begin{aligned} f_{MS} &= \frac{f_{CLK} \cdot v}{2^{pd} \cdot 2048 \cdot 32} \\ &= \frac{16 \text{ MHz} \cdot 250}{2^3 \cdot 2048 \cdot 32} = 7629,4 \text{ Hz} \end{aligned}$$

Die Vollschrift-Frequenz f_{VS} ergibt sich aus der Mikroschrittfrequenz und der eingestellten Schrittauflösung der Steuerkarte:

$$\begin{aligned} f_{VS} &= \frac{f_{MS}}{S_{MS}} \\ &= \frac{7629,4 \text{ Hz}}{16} = 476,84 \text{ Hz} \end{aligned}$$

Zur Berechnung der Drehzahl n wird die Vollschrift-Frequenz durch die Schrittauflösung des Motors dividiert:

$$\begin{aligned} n &= \frac{f_{VS}}{S_{MOTOR}} \\ &= \frac{476,84 \text{ Hz}}{200} = 2,384 \text{ s}^{-1} = 143,04 \text{ min}^{-1} \end{aligned}$$

3.8 Endlagenschalter

Jede Achse verfügt über je einen Taster für die positive und negative Endlage. Die Betätiger der Taster an den Achsenschiebern sind verstellbar, daher können die Grenzen des Fahrbereichs feinjustiert werden. Bei Betätigung des Schalters wird im Steuerungsmodul ein Hardware-Stopp ausgelöst.

3.9 XY-Schiff

Das XY-Schiff ist der Träger der Messsonde. Es verfügt über keine direkte Fixierung, sondern „fährt“ durch die Schub- und Zugsbewegung der beiden Achsenschieber auf der XY-Ebene (daher auch die Bezeichnung „Schiff“).

Aufgrund der Reibung an den Grenzflächen zu den Schiebern bzw. zum Anlagenboden sind die Flächen entsprechend geschmiert.

3.10 Z-Rahmen

Der Rahmen für die Befestigung der Z-Achse besteht aus vier verschraubten Kunststoffteilen sowie der Haltevorrichtung für den Z- Schrittmotor. Am unteren Teil des Rahmens ist ein Aluminiumblock angeschraubt, in welchem das Kugellager der Z-Spindel eingepresst ist. Die Höhe des Rahmens und respektive auch der Fahrbereich der Z-Achse sind entsprechend der geforderten Applikation im binauralen Hörsystem gewählt.

3.11 Z-Spindel

Die Z-Spindel wurde aufgrund der Anforderungen hinsichtlich Form und Abmessungen speziell für die Anlage angefertigt. Da die Spindel der ersten Anlage kürzer ist als die des Nachbaus, unterscheiden sich die Positionen der Z-Lager am unteren Teil des Z-Rahmens und somit auch die Fahrbereiche beider Anlagen (s. Abbildung 12). Auf die Messungen hat dies jedoch keinen Einfluss, da der Mindestabstand zwischen Referenzgerät und Messsonde noch über der unteren Z-Endlage liegt.

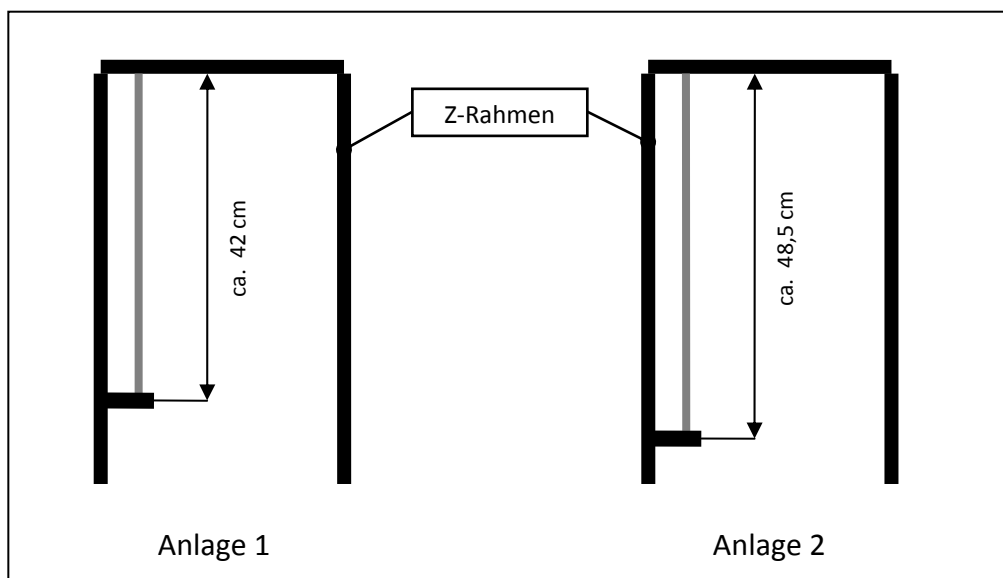


Abbildung 12: unterschiedliche Spindellängen von Anlage 1 und 2

Anders als bei X- und Y-Achse ist die Spindel mittels einer Verdrehsicherung fest mit der rotierenden Motorbuchse verbunden und fungiert somit als Motorwelle. Am anderen Ende befindet sich ein entsprechendes Kugellager. Die Linearbewegung wird daher nicht von der Spindel, sondern vom Z-Träger ausgeführt.

3.12 Z-Träger

3.12.1 Aufbau und Funktion

Der Z-Träger wirkt zusammen mit Spindel und Motor als ein Gleitgewindetrieb. Durch die Führung in den Gewindebuchsen wandelt dieser die Rotation der Spindel in eine Längsbewegung entlang der Z-Achse. Um die Reibverluste zwischen Spindel und Gewindebuchse am Träger zu reduzieren, sind diese aus Teflon gefertigt. Auf der gegenüberliegenden Seite wird die Führung parallel zur Achse durch eine Nut im Z-Rahmen realisiert. Analog zur X- und Y-Achse sind die Betätiger der Endschalter auch am Z-Träger verstellbar eingeschraubt (s. Abbildung 13).

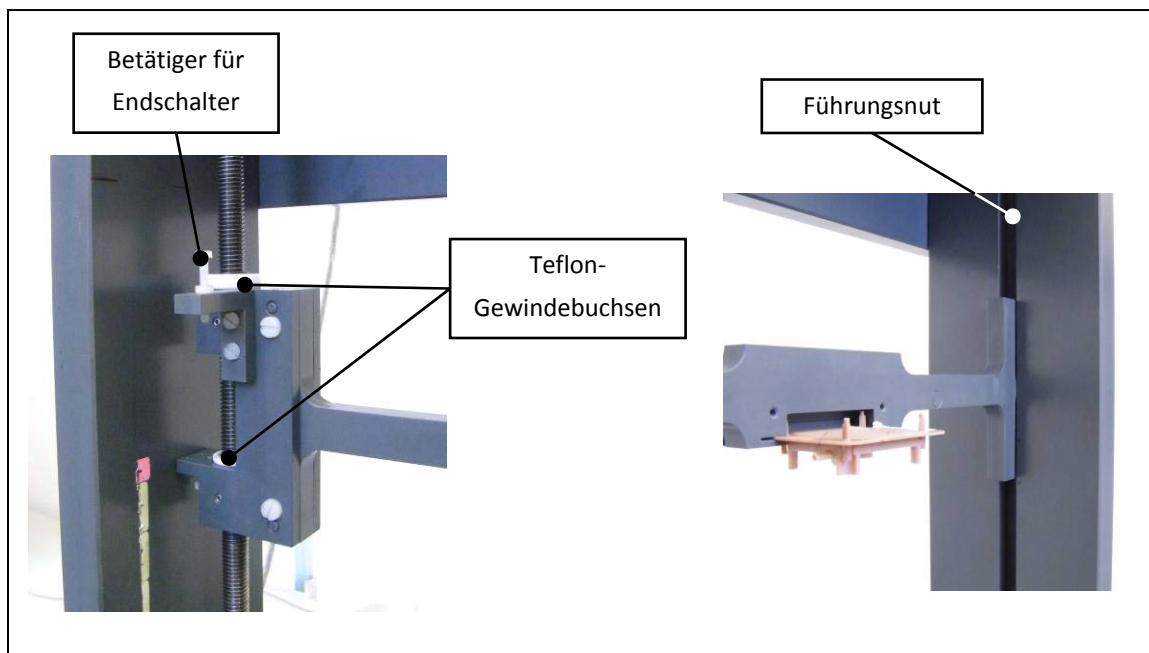


Abbildung 13: Z-Träger

3.13 Z-Schrittmotor

3.13.1 Aufbau und Funktion

Die Z-Achse verfügt über den gleichen Schrittmotor wie X- und Y-Achse. Der Unterschied liegt hierbei nur in der Fixierungsart der Spindel. Während bei X- und Y-Achse die Spindel

axial geführt wird, ist die breitere Z-Spindel an ihrem Ende in die Gewindebuchse eingeschraubt und somit auch axial fixiert.

3.13.2 Umrechnung der eingestellten Drehzahl

Die Berechnung der Drehzahl erfolgt analog zu 3.7.2. Aufgrund der längeren Spindel und dem größeren Fahrbereich im Vergleich zu X- und Y-Achse ist der Geschwindigkeitswert höher eingestellt:

Tabelle 7: Parameter zur Berechnung der Z-Drehzahl

Parameter	Formelzeichen	Wert
TMCM-Parameter		
Taktfrequenz	f_{CLK}	16 MHz
Geschwindigkeit	v	500 (gewählt)
Pulse Divisor	pd	3 (gewählt)
Mikroschrittauflösung	S_{MS}	16 Mikroschritte/Vollschritt
Motorparameter		
Schrittauflösung	S_{MOTOR}	200 Vollschritte/Umdrehung

Damit ergibt sich für die Berechnung nach [4]:

(1) Mikroschritt-Frequenz

$$\begin{aligned}
 f_{MS} &= \frac{f_{CLK} \cdot v}{2^{pd} \cdot 2048 \cdot 32} \\
 &= \frac{16 \text{ MHz} \cdot 500}{2^3 \cdot 2048 \cdot 32} = 15258,8 \text{ Hz}
 \end{aligned}$$

(2) Vollschritt-Frequenz

$$\begin{aligned}
 f_{VS} &= \frac{f_{MS}}{S_{MS}} \\
 &= \frac{15258,8 \text{ Hz}}{16} = 953,68 \text{ Hz}
 \end{aligned}$$

(2) Drehzahl

$$\begin{aligned} n &= \frac{f_{VS} \cdot 60}{S_{MOTOR}} \\ &= \frac{953,68 \text{ Hz} \cdot 60}{200} = 286,1 \text{ min}^{-1} \end{aligned}$$

3.13.3 Umrechnung des Haltemomentes

Aus dem Datenblatt des bisher verwendeten NANOTEC-Motors geht für das maximale Haltemoment ein Wert von 1100 g-cm hervor [7].

Dies bedeutet dass in 1 cm Entfernung von der Drehachse ein Gewicht von 1100 g gehalten werden kann. Rechnerisch ergibt sich das Haltemoment daher wie folgt [19]:

$$\begin{aligned} M_H &= m \cdot g \cdot l = 1100 \text{ g} \cdot 9,81 \frac{\text{m}}{\text{s}^2} \cdot 1 \text{ cm} \\ &= 10,8 \text{ Ncm} \end{aligned}$$

m	Masse [g]
g	Normalfallbeschleunigung [m/s ²]
l	Abstand zur Drehachse [cm]

Das berechnete Haltemoment wird jedoch nur bei maximalem Phasenstrom ausgeübt. Da der Netzteil-Ausgangstrom (s. 3.3.4) unter dem maximalen Phasenstrom liegt, ist das maximal aufgebrachte Drehmoment des Motors somit geringer. Da der Hersteller keine Motorkennlinie zur Verfügung stellen konnte, ist das reell ausgeübte Moment des Motors bei der eingestellten Drehzahl nicht weiter präzisierbar.

3.14 Referenzgerät

Das Referenzgerät ist standardmäßig als das rechte Gerät definiert und enthält alle wesentlichen Komponenten eines HdO oder IdO. Es dient ausschließlich zur Durchführung des Pingpong-Tests bzw. dessen Ergebnisanzeige durch verschiedene Tonausgaben. Die Sendespule (Wireless-Antenne) ist in ausreichendem Abstand zu den übrigen Komponenten platziert, um ein störungsfreies Sende- und Empfangsverhalten zu gewährleisten.

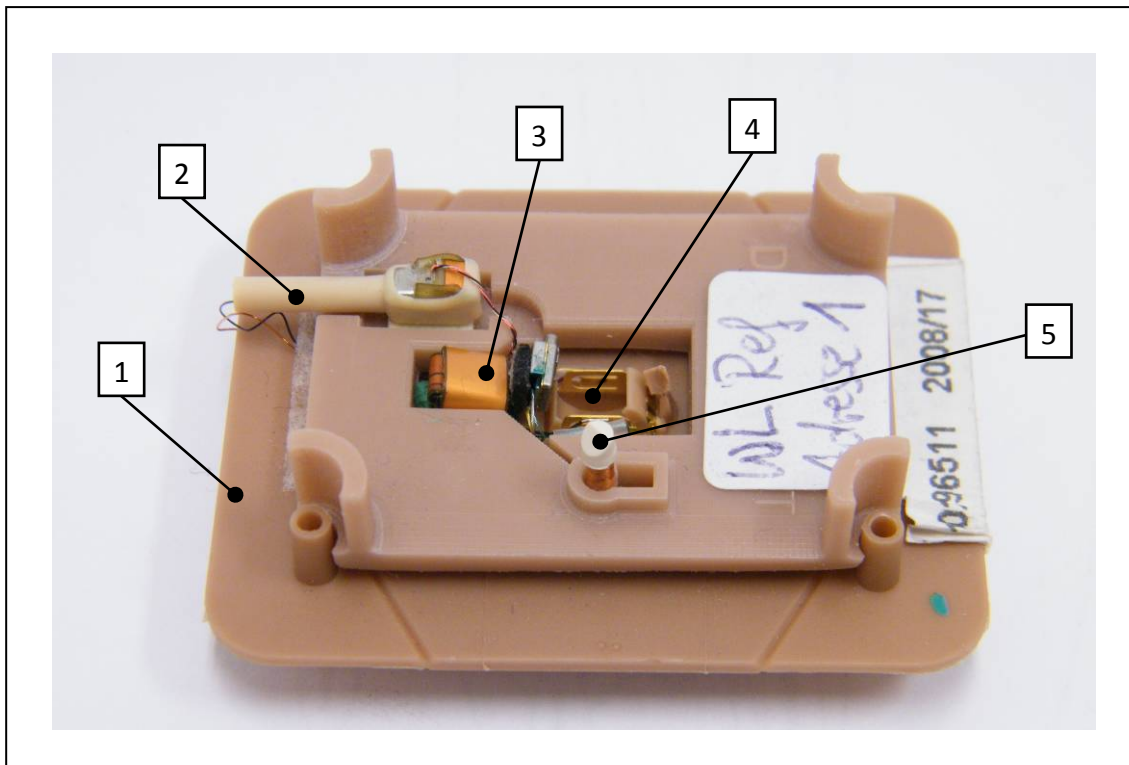


Abbildung 14: Referenzgerät

Tabelle 8: Komponenten Referenzgerät

Nr.	Bezeichnung
1	Faceplate (Träger)
2	Hörer (Tonausgabe)
3	Verarbeitungselektronik (Hybrid)
4	Batteriefach
5	Wireless-Antenne (Sendespule)

3.15 Messsonde

Hauptbestandteile der Messsonde sind die Messspule und deren Elektronik (Hybrid). Die Spule ist über ein Koaxialkabel mit der auf dem Faceplate befindlichen Elektronik verbunden (s. Abbildung 15). Da die Spule genau im Mittelpunkt des Trägers befestigt ist, bildet diese im Messaufbau immer den Schnittpunkt von X- und Y-Achse. Damit die Position der Messspule am Testobjekt erkennbar ist, wurde der Träger aus Acrylglas gefertigt.

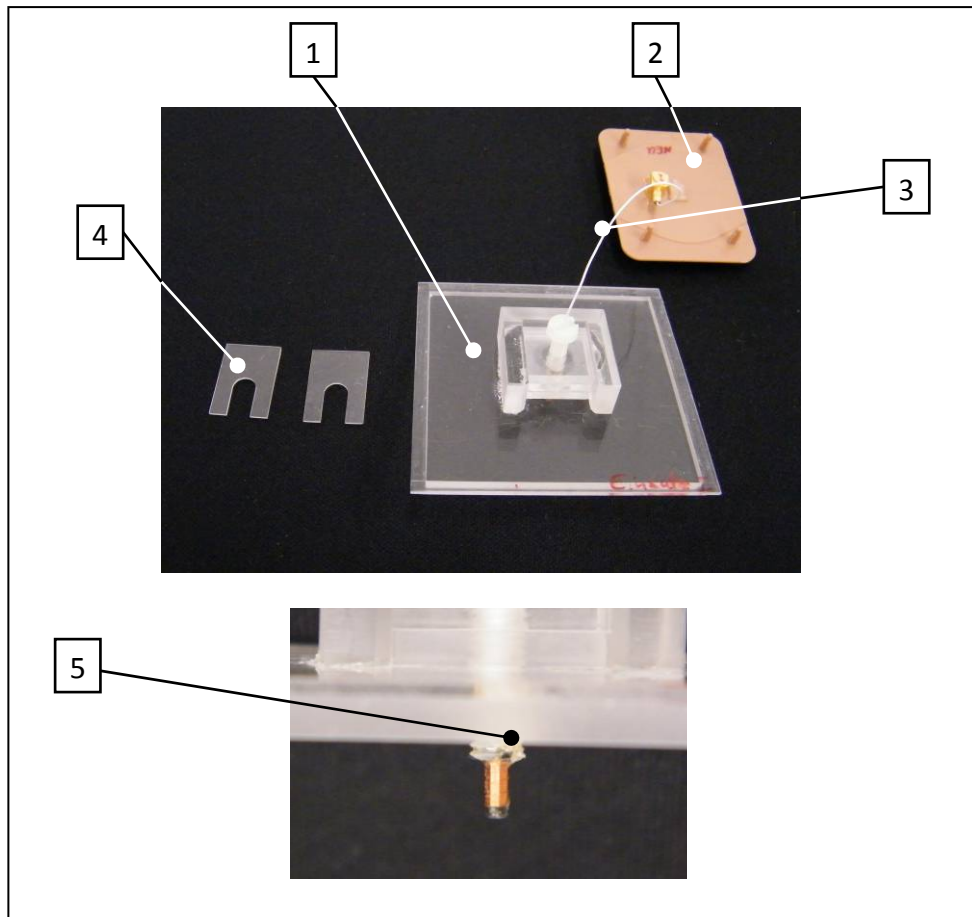


Abbildung 15: Messsonde

Tabelle 9: Komponenten Messsonde

Nr.	Bezeichnung
1	Träger
2	Faceplate mit Elektronik
3	Koaxialkabel
4	Einschubplättchen
5	Messspule

Die Höhe der Messspule auf der Objekt-Z-Achse (Eintauchtiefe) kann mit Hilfe von Einschubplättchen millimeterweise eingestellt werden (s. Abbildung 16).

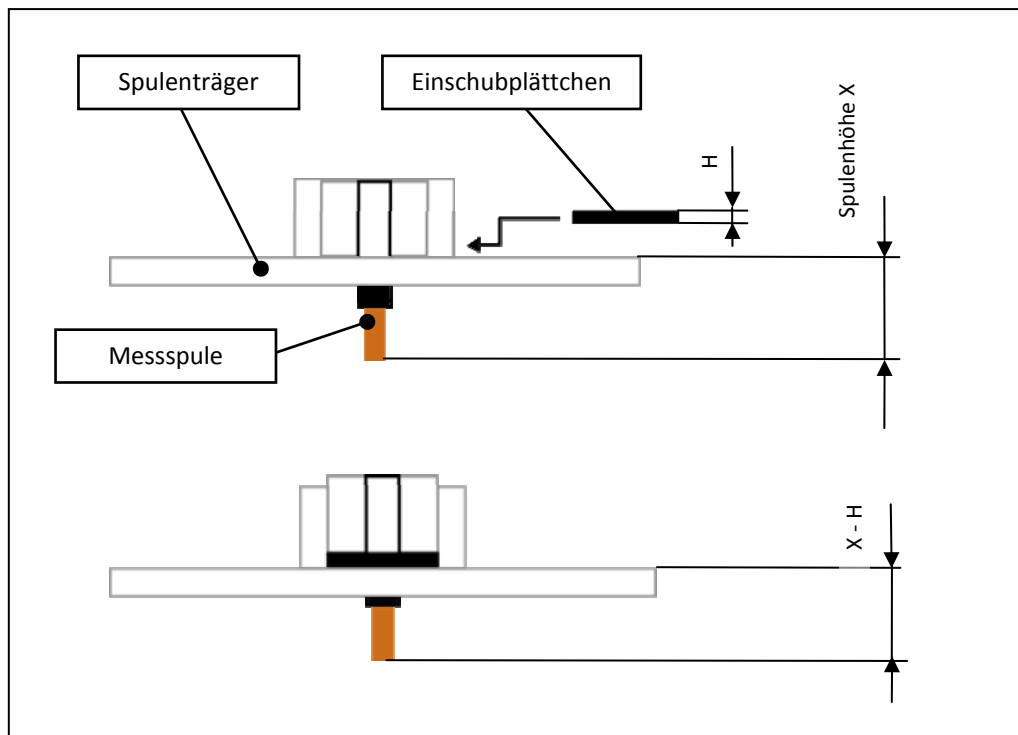


Abbildung 16: Erhöhung des Spulenabstandes

3.16 Standalone Wireless Tester

Der Standalone Wireless Tester (SWT) ist eine speziell programmierte Benutzerfernbedienung deren Funktionsumfang für die Entwicklung und Untersuchung der Hörgeräte erweitert wurde. Die Bezeichnung „Standalone“ beschreibt hierbei die Fähigkeit des Gerätes, die Hörgeräte selbstständig (ohne PC-Unterstützung) programmieren zu können.

Bei den Messungen wird der SWT hauptsächlich zum Trimmen der beiden Hörgeräte genutzt und kann zusätzlich noch zur Batterieüberwachung verwendet werden.

Trimmroutine und Statusabfrage erfolgen per Tastendruck. Der Status wird anschließend im Display angezeigt.



Abbildung 17: Standalone Wireless Tester

3.17 Testfaceplate

Analog zu Referenzgerät und Messsonde sind auch die Testobjekte auf Faceplates untergebracht. Alle Testobjekte (mit Ausnahme von Batterien) werden von einer Hörgeräteelektronik (Hybrid) angesteuert, die sich ebenfalls auf dem Faceplate befindet.

Für Objekte mit definierten Abmessungen wie Batterien oder Hörer existieren zusätzlich spezielle Träger, auf denen die Bauteile fixiert werden können. Die Zusatzträger werden dabei direkt neben dem Testfaceplate platziert (s. Abbildung 18).

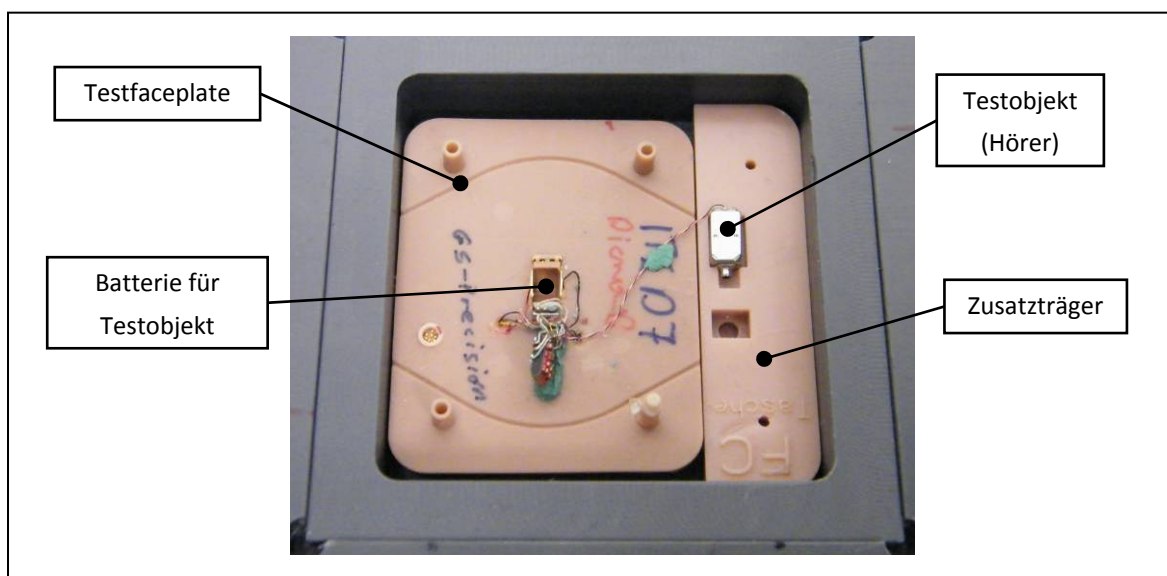


Abbildung 18: Testfaceplate mit Hörer als Testobjekt

4 Stand der Technik - Software

Die Basis für die Messsoftware bildet die Programmierplattform MATLAB. In der Siemens Audiologischen Technik GmbH ist diese eine weit verbreitete Grundlage für die Erstellung von Konfigurationstools sowie Mess- und Auswertefunktionen mit grafischen Benutzeroberflächen (GUI's).

Die aktuelle GUI baut sich wie folgt auf:

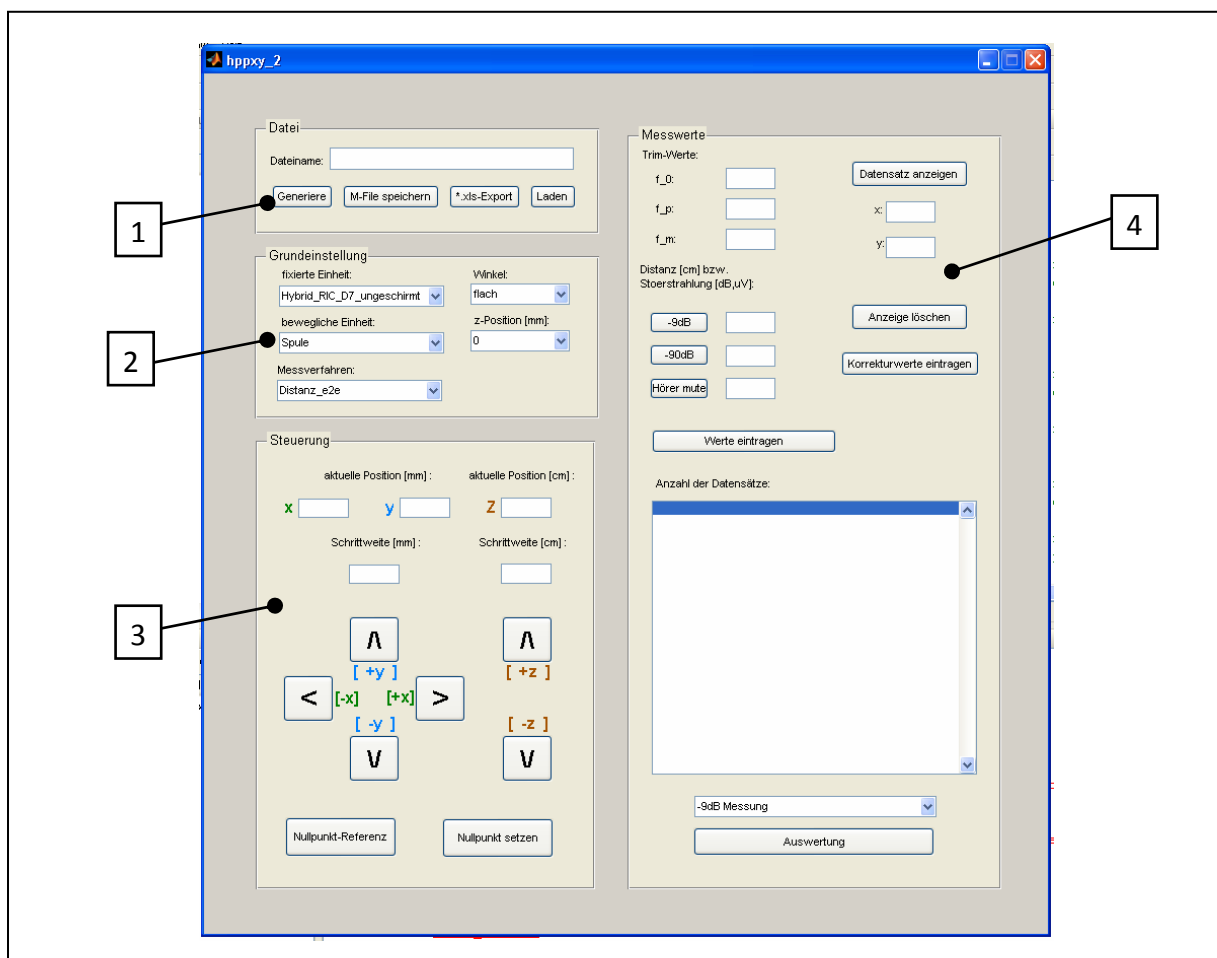


Abbildung 19: Benutzeroberfläche

Tabelle 10: Komponenten der GUI

Nr.	Beschreibung
1	Speichern der Daten als mat- oder csv-Datei, Öffnen vorhandener Daten
2	Messeinstellungen; die eingestellten Werte werden beim Speichern dem Dateinamen hinzugefügt
3	schrittweise Steuerung der Motoren und Anzeige der aktuellen Position
4	Eintragen der Messwerte; Anzeige der Daten in Diagramm

Die vorhandene Programmierung wird hier nicht weiter betrachtet, da für die Automatisierung eine von Grund auf neue Anlagensoftware erstellt wird.

5 Anforderungen und aktueller Stand im Vergleich

Verallgemeinert kann gesagt werden, dass der gesamte Messablauf (s. 2.2.3) selbstständig ablaufen soll. Der Nutzer soll lediglich die Vorkonfiguration vornehmen und nach Abschluss einer Messung die gespeicherten Daten auswerten können. Zur Verdeutlichung der Anforderungen wird die schrittweise Automatisierung wie folgt dargestellt:

5.1 Hardware

5.1.1 Konfiguration

Gefordert wird ein fester Satz an Grundeinstellungen für die Anlage, der gespeichert und bei eventuellen Ausfällen neu geladen werden kann. Dies umfasst:

- Motoreinstellungen (Geschwindigkeit, Strom, ...)
- Schnittstellenparameter (COM-Port, Baudrate, ...) und
- messrelevante Einstellungen (Startzeit, Flags, etc.).

Bisher sind nur die Motoreinstellungen ausfallsicher im EEPROM der Schrittmotorkarte gespeichert. Die Schnittstellenkonfiguration muss „von Hand“ vorgenommen werden. Einen kompletten Parametersatz für die Anlage gibt es nicht.

5.1.2 Feineinstellung der Messspule

Gefordert wird eine stufenlose, definierte Höheneinstellung der Messspule für die exakte Annäherung an das Testfaceplate.

Bisher kann die Messspule in ihrem Träger nur millimeterweise anhand von Kunststoffplättchen höhenverstellt werden.

5.1.3 Z-Achse

Gefordert wird eine höhere Geschwindigkeit zur Positionierung des Referenzgerätes. Da die Z-Achse einen weitaus größeren Fahrbereich als X- und Y-Achse umfasst und generell längere Wege zurücklegt, soll hier ein leistungstärkerer Motor zum Einsatz kommen.

Bisher entspricht der Z-Motor denen von X- und Y-Achse. Um den kompletten Fahrbereich abzufahren, benötigt die Z-Achse damit mehr als doppelt so viel Zeit wie X- oder Y-Achse.

5.2 Messungen

5.2.1 Definition der Messpunkte

Gefordert wird eine vereinfachte Definition des Messbereiches am Testobjekt. Dafür soll es mehrere Möglichkeiten geben wie z.B.:

- die grafische (flächenmäßige) Definition des Messbereiches
- das Festlegen einzelner Messpunkte oder
- die Übernahme von Messpunkten/ -bereichen aus der aktuellen Motorposition

Bisher wird jeder Messpunkt, und somit der gesamte Messbereich, einzeln angefahren. Einen direkt vordefinierten Bereich gibt es dabei nicht.

5.2.2 Pingpong-Test

Gefordert wird der eigenständige, automatische Verbindungstest zwischen Referenzgerät und Messsonde. Dazu müssen die Hörgeräte von der Software, d.h. vom PC aus, angesteuert werden können.

Bisher muss der Benutzer das Testergebnis akustisch anhand der Tonausgabe des Referenzgerätes bewerten. Da eine automatisierte Auswertung dieser Töne jedoch schwer zu realisieren ist, soll dieser Test nun durch eine Software-Statusabfrage über den Wireless Programmer ersetzt werden.

5.2.3 Automatische Trimmung

Gefordert wird die automatische Frequenzabstimmung (Trimmen) der Hörgeräte in regelmäßigen Abständen während des laufenden Messzyklus.

Bisher erfolgt das Trimmen manuell per Tastendruck am Standalone Wireless Tester.

5.2.4 Messablauf

Gefordert wird eine automatisierter Messzyklus (entsprechend 2.2.3). Die Anlage soll die vordefinierten XY-Messpunkte selbstständig anfahren können und dort den Pingpong-Test durchführen. In Abhängigkeit von dessen Ergebnis soll anschließend der Abstand zwischen Referenzgerät und Messsonde, d.h. die Z-Position, erhöht oder verringert werden. Nachdem der maximale Z-Abstand erreicht ist, sollen der X-, Y- und Z-Wert gespeichert werden. Es soll ein autarker Messbetrieb der Anlage gewährleistet sein, der die ständige Steuerung und Überwachung durch den Benutzer erspart.

Bisher muss der Benutzer jeden XY-Punkt manuell über die Fahrschaltflächen des Messprogramms anfahren. Dementsprechend muss nach dem Pingpong-Test auch die Z-Achse manuell über die Programmbefehle verfahren. Die Anlage ist daher nur als halbautomatisch zu betrachten.

5.2.5 Datenspeicherung und -auswertung

Gefordert wird die automatische Sicherung der Messdaten. Dies soll nach jedem Messpunkt erfolgen. Für die erleichterte Auswertung sollen zusätzlich verschiedene Speicherformate wählbar sein (z.B. *.xml, *.csv). Gewünscht wird zusätzlich ein Auswertetool, das die Daten aussagekräftig veranschaulichen kann.

Bisher müssen die XYZ-Werte nach jedem Messpunkt manuell gespeichert werden. Die Speicherung beinhaltet lediglich den Datenexport in Microsoft Excel. Als Auswerteprogramm wurde hierbei nur die MATLAB-interne Darstellung als Diagramm genutzt.

5.3 PC-Benutzeroberfläche

5.3.1 Bedienung

Gefordert wird leichtere Bedienung und Übersichtlichkeit. Die Oberfläche soll alle wichtigen Funktionen enthalten und zudem optisch ansprechend sein.

Bisher enthält die GUI im Wesentlichen nur Funktionen zur schrittweisen Bewegung der Achsen sowie zum Datenspeichern.

5.3.2 Fahrfunktionen

Gefordert wird dass die Achsen auf verschiedene Arten unkompliziert positioniert werden können, z.B. durch relative oder absolute Positionierung oder durch freies Verfahren über Pfeiltasten.

Bisher können in der GUI die Achsen jeweils nur relativ verfahren werden. Als Grundeinheit gilt dabei Millimeter für die X- und Y-Achse und Zentimeter für die Z-Achse.

5.3.3 Statusüberwachung

Gefordert wird dass die Anlage im laufenden Messbetrieb überwacht werden kann. Die GUI soll z.B. über den Gesamtfortschritt und die verbleibende Messzeit informieren. Des Weiteren ist im Positionierbetrieb von Bedeutung, dass die aktuelle Position angezeigt wird. Gewünscht wird zudem die Überwachung der Batteriekapazität von Referenzgerät und Messsonde.

Bisher enthält die GUI keinerlei Informationen über den Gerätestatus. Die aktuelle Position wird nur indirekt über die Relation zum Startpunkt angezeigt.

5.3.4 Einstellmöglichkeiten

Gefordert wird die Möglichkeit, Anlageneinstellungen unkompliziert verändern zu können. In Anlehnung an den Standard in technischen Benutzeroberflächen, soll der Nutzer dabei über ein Einstellungsfenster alle Parameter verändern, laden und speichern können (Vgl. auch 5.1.1).

Bisher existieren keine Einstellmöglichkeiten in der GUI. Motoreinstellungen müssen über die Parametriersoftware (TMCL-IDE) der Schrittmotorkarte vorgenommen werden.

6 Umsetzung der Anforderungen

6.1 Hardware

6.1.1 Stufenlose Höheneinstellung der Messspule

Um die Höhe der Messspule am Testobjekt variabel einstellen zu können, wurde eine Trägervorrichtung angefertigt, die dies über zwei Zahnräder stufenlos ermöglicht. Als Material wurde aufgrund der Vermeidung passiver Störungen wieder Kunststoff gewählt.

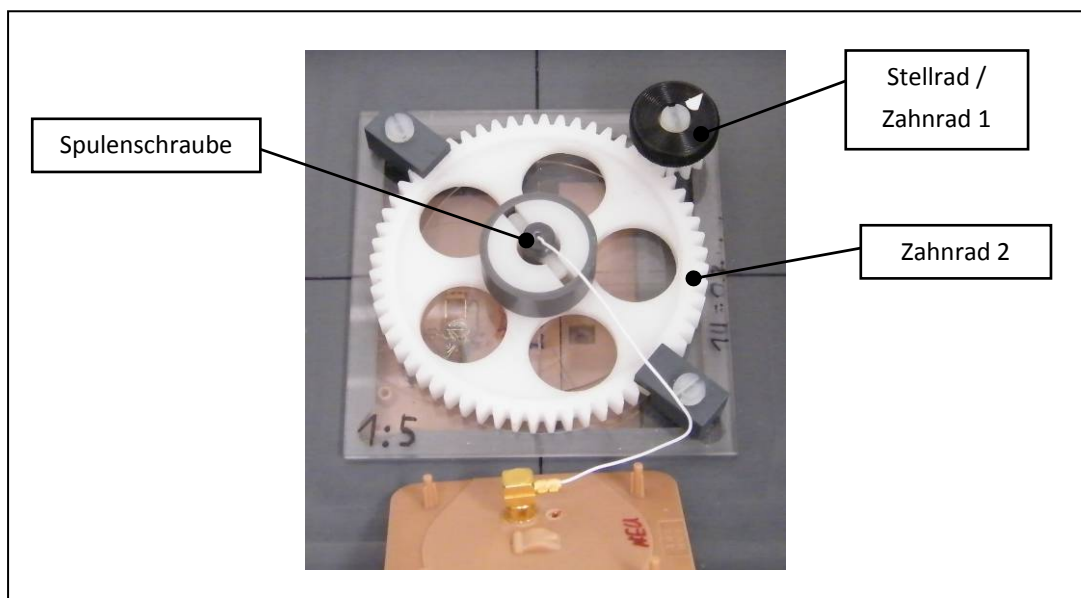


Abbildung 20: Trägervorrichtung der Messspule

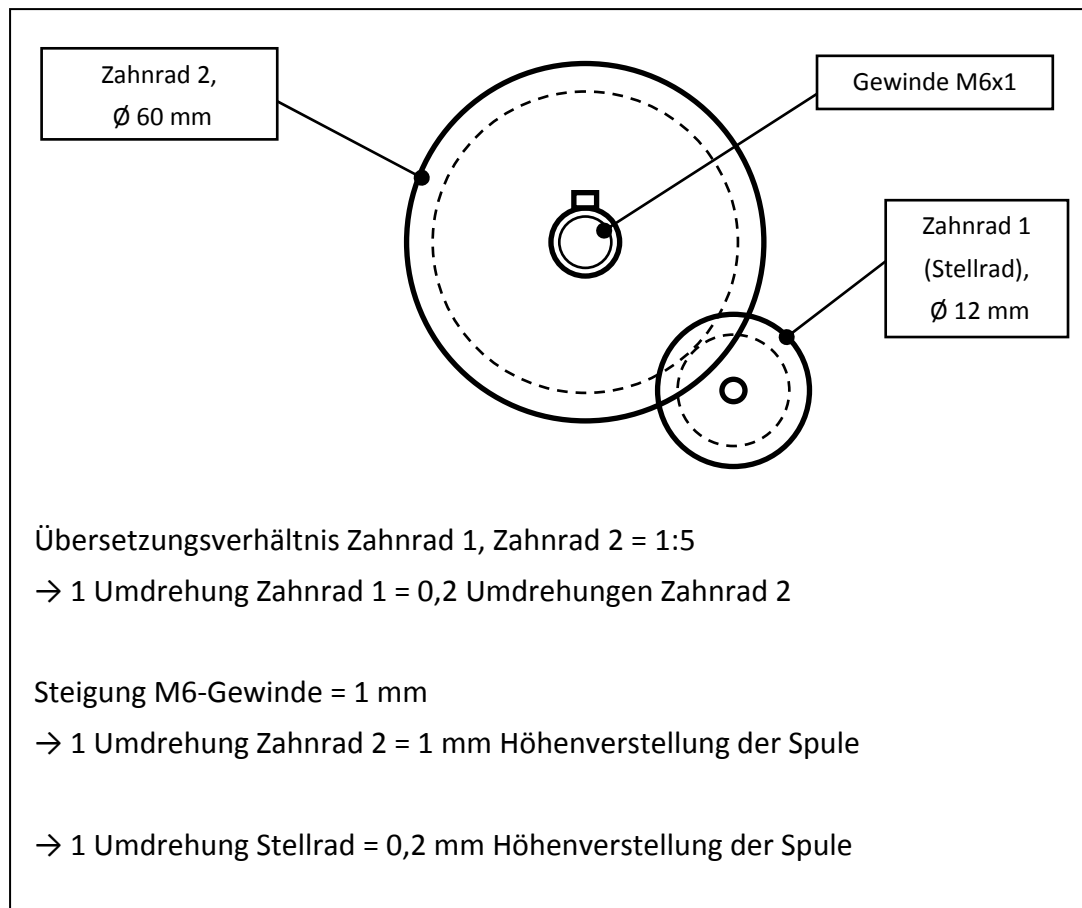


Abbildung 21: Höhenverstellung der Spule

Die beiden Zahnräder besitzen ein Übersetzungsverhältnis von 1:5. Die Spule ist mit einem M6-Gewinde mit einer Steigung von 1 mm in der Mitte der Vorrichtung eingeschraubt und fest mit Zahnrad 2 verbunden. Durch das Übersetzungsverhältnis von 1:5 bewirkt eine Umdrehung am Stellrad (Zahnrad 1) eine Fünftelumdrehung an Zahnrad 2. Aufgrund der Steigung des M6-Gewindes bewirkt eine Umdrehung am Zahnrad 2 somit eine Höhenverstellung von 1 mm.

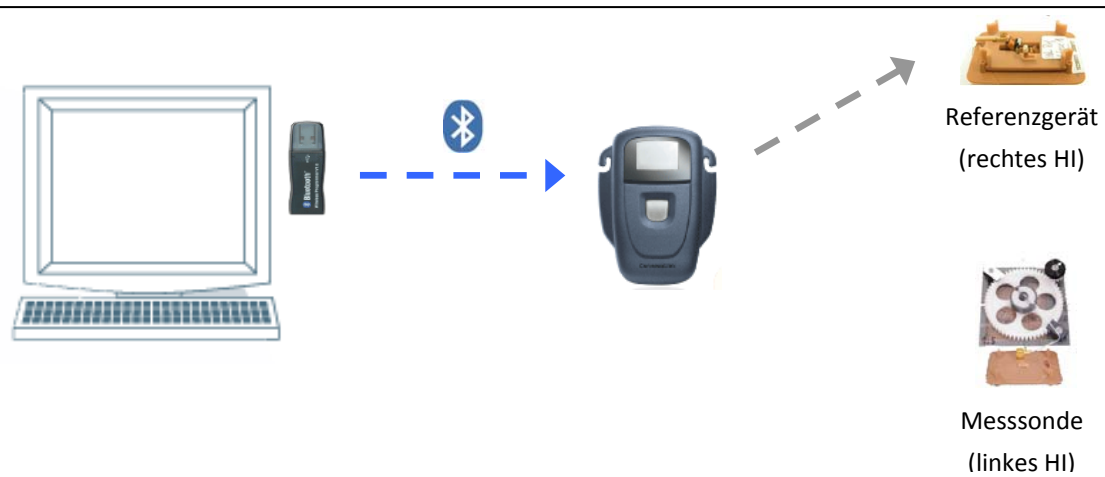
6.1.2 Einführung des Wireless Programmers

Der Wireless Programmer (WLP) ist ähnlich dem SWT ein spezielles Handgerät, das zum Programmieren der Hörgeräte verwendet wird. Die Kommandos werden hierbei mit Hilfe eines Bluetooth-USB-Adapters vom PC aus an die Hörgeräte gesendet (s. Abbildung 22).



Abbildung 22: WLP und Adapter

Zweck des WLP ist es, zum Einen den akustischen Pingpong-Test durch verarbeitbare Softwaresignale zu ersetzen, und zum Anderen die Trimmung der Hörgeräte automatisch initiieren zu können. Die Kommunikation zwischen PC und Hörgerät erfolgt nach dem folgenden Schema:



Über den Bluetooth-Adapter wird ein Datenframe an den WLP übermittelt. Dieser sendet die Daten über Funk zum Referenzgerät.



Das Referenzgerät versucht die Verbindung mit einem gegenüberliegenden linken Gerät herzustellen und die Daten weiterzugeben. Bei erfolgreicher Erkennung sendet das Referenzgerät eine definierte Bytefolge mit Informationen beider Geräte an den WLP zurück. Die Bytefolge wird an den PC übermittelt und kann dort entsprechend ausgewertet werden.

(Fortsetzung nächste Seite)

(Fortsetzung)

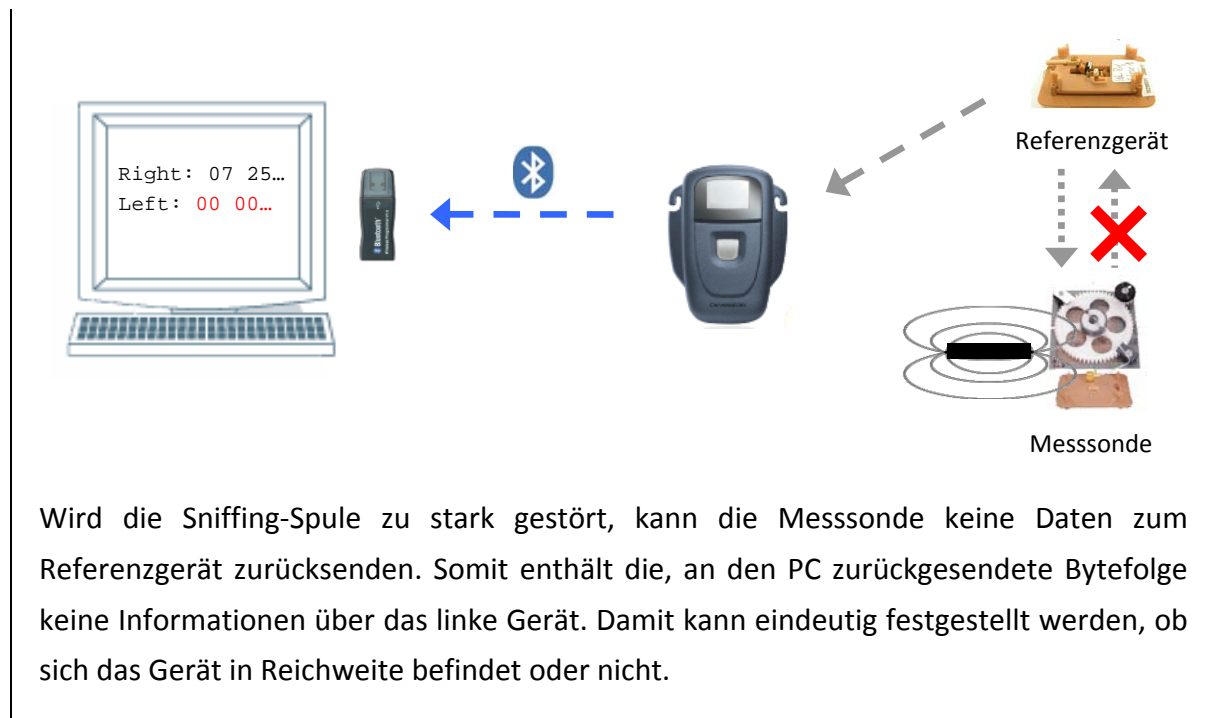


Abbildung 23: Kommunikationssystematik beim Auslesen der Hörgeräte

6.1.3 Netzteil

Das bisher verwendete Netzteil (VOLTCRAFT PA300) (s. Tabelle 5) wurde aufgrund der geringen Leistung durch ein stärkeres (VOLTCRAFT SNG24) ersetzt. Tabelle 11 zeigt die neuen Daten des neuen Netzteils im Vergleich zu den bisher verwendeten.

Tabelle 11: Vergleich der Netzteilangaben [6], [9]

	VOLTCRAFT PA300	VOLTCRAFT SNG24
Ausgangsspannung	12 V	12 V
Ausgangsstrom	0,3 A	2 A
Ausgangsleistung	3,6 W	24 W

Analog zur Rechnung in 3.3.4 berechnet sich der Phasenstrom pro Motor wie folgt:

$$I_{PH} = \frac{I_A}{N}$$

N Phasenanzahl

$$= \frac{2 \text{ A}}{2} = 1 \text{ A}$$

Die Schrittmotoren besitzen einen maximalen Phasenstrom von 0,5 A [7]. Die Netzteile sind somit ausreichend. Begrenzt wird der Phasenstrom durch die entsprechende Einstellung im TCM-Modul.

6.1.4 Z-Schrittmotor

6.1.4.1 Dimensionierung

Um den Motor vorab dimensionieren zu können, wird zunächst das erforderliche Drehmoment M_z [Nm] zur Bewegung des Z-Trägers berechnet [10].

$$M_z = \frac{F}{2\pi \cdot \eta} \cdot \frac{P}{1000}$$

F Axialkraft an der Spindel [N]
 P Gewindesteigung [mm]
 η Wirkungsgrad des Spindeltriebes

$$= \frac{F \cdot P}{2000 \cdot \pi \cdot \eta}$$

Für eine überschlägige Berechnung kann die Axialkraft F [N] aus der Gewichtskraft des Z-Trägers ermittelt werden. Das Gewicht des Z-Trägers beträgt ca. 130 g.

Somit gilt für die Axialkraft:

$$F = m \cdot g$$

m Masse des Z-Trägers [g]
 g Normalfallbeschleunigung [m/s^2]

$$= 0.013 \text{ kg} \cdot 9,81 \text{ m/s}^2$$
$$\approx 1,3 \text{ N}$$

Der Wirkungsgrad η des Spindeltriebes wird berechnet mit [10]:

$$\eta = \frac{\tan \alpha}{\tan(\alpha + \gamma)}$$

α Steigungswinkel des Gewindes [°]
 γ Reibungswinkel [°]

Der Steigungswinkel α berechnet sich wie folgt [10]:

$$\alpha = \arctan \frac{P}{d_2 \cdot \pi}$$

P Gewindesteigung [mm]
 d_2 Flankendurchmesser des Gewindes [mm]

Der Flankendurchmesser eines M10-Gewindes mit einer Steigung von 1mm beträgt laut [14] 9,35 mm.

Somit gilt für α :

$$\alpha = \arctan \frac{1 \text{ mm}}{9,35 \text{ mm} \cdot \pi} = 1,95^\circ$$

Der Reibungswinkel γ wird mit der folgenden Formel berechnet [12]:

$$\gamma = \arctan \frac{\mu}{\cos\left(\frac{\beta}{2}\right)}$$

μ Reibungskoeffizient
 β Flankenwinkel [°]

Der Reibungskoeffizient der Paarung Stahl-PVC hat laut [13] einen Wert von ca. 0,6

Der Flankenwinkel β beträgt bei metrischen ISO-Feingewinden 60° [11].

Somit ergibt sich für den Reibungswinkel:

$$\gamma = \arctan \frac{0,6}{\cos\left(\frac{60^\circ}{2}\right)} = 34,72^\circ$$

Der Wirkungsgrad beträgt demnach:

$$\eta = \frac{\tan 1,95^\circ}{\tan(1,95^\circ + 34,72^\circ)} = 0,046$$

Damit kann M_z berechnet werden:

$$M_z = \frac{1,3 \text{ N} \cdot 1 \text{ mm}}{2000 \cdot \pi \cdot 0,046} = 0,0045 \text{ Nm} = 4,5 \text{ Ncm}$$

Das errechnete Drehmoment dient hier nur als Richtwert, da die Gewichtskraft des Trägers nicht axial an der Spindel wirkt sondern als Moment in dessen Schwerpunkt. Dieses Moment ruft zusätzliche Reibung hervor, ist jedoch nur mit hohem Aufwand berechenbar. Für die

weitere Dimensionierung wird daher ein Sicherheitsfaktor von 2 hinzugerechnet, somit ergibt sich für das erforderliche Drehmoment:

$$M_z = 2 \cdot 4,5 \text{ Ncm} = 9 \text{ Ncm}$$

Das aufzubringende Drehmoment liegt damit, auch unter Verwendung der neuen Netzteile, noch unter dem maximalen Haltemoment des bisher verwendeten Schrittmotors (s. 3.13.3 bzw. 6.1.3).

Ziel bei der Z-Achse war jedoch eine möglichst hohe Drehzahl bzw. Verfahrgeschwindigkeit zu erreichen. Aufgrund der Tatsache, dass der bisherige Z-Motor bei höherer Drehzahl ein weitaus geringeres Drehmoment als das angegebene Haltemoment ausübt, wurde dieser durch den stärkeren Motor (Typ TRINAMIC QSH5718, s. Abbildung 24) ersetzt.

Im Gegensatz zum bisher verwendeten Schrittmotor verfügt dieser über eine fest integrierte Motorwelle, an die die Z-Spindel mittels einer Verdrehsicherung befestigt wird.



Abbildung 24: TRINAMIC QSH5718-51-28-101

Tabelle 12 zeigt die Daten der beiden Motoren im Vergleich.

Tabelle 12: Vergleich der Z-Schrittmotoren alt und neu [7], [15]

Bez./Parameter	NANOTEC L4018	TRINAMIC QSH5718
Nennspannung	4,5 V	2,3 V
max. Phasenstrom	0,5 A	2,8 A
Schrittauflösung	200 Schritte/U	200 Schritte/U
Haltemoment	0,0108 Nm	1,01 Nm

6.1.4.2 Einstellung der Drehzahl

Aus der Drehmomentkennlinie des Motors (s. Abbildung 25) geht hervor, dass das maximale Drehmoment bei einer Schrittfrequenz von ca. 1800 Hz anliegt.

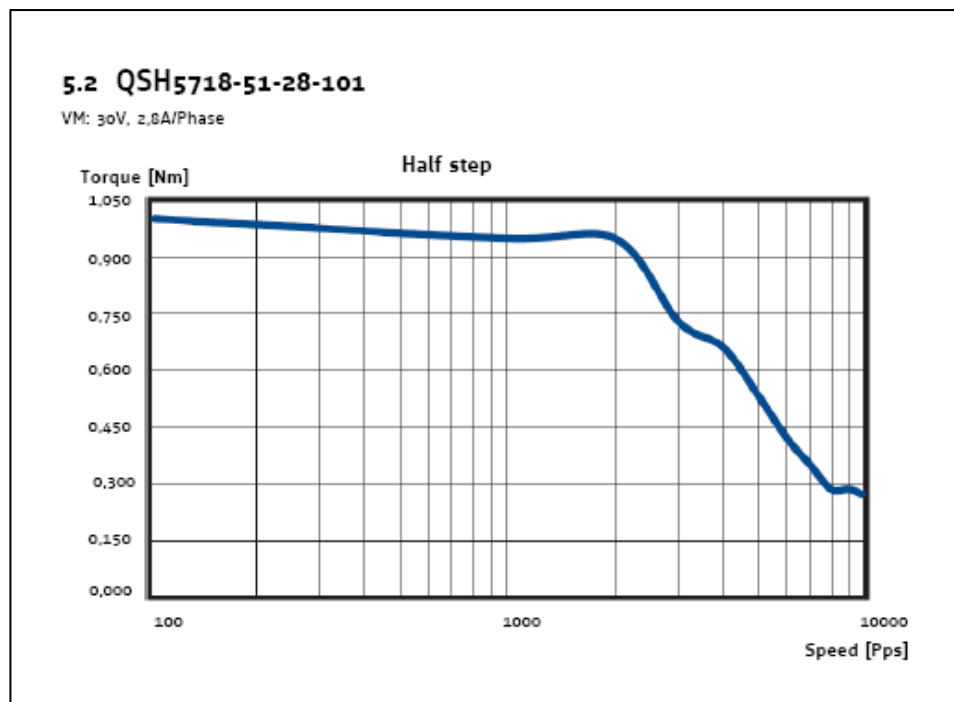


Abbildung 25: Motorkennlinie Trinamic QSH5718-51-28-101 [15]

Anhand der Formel aus 3.7.2 kann damit die Drehzahl errechnet werden:

Tabelle 13: Parameter zur Drehzahlberechnung

Parameter	Formelzeichen	Wert
TMCM-Parameter		
Schrittfrequenz	f_s	1800 Hz
Motorparameter		
Schrittauflösung	S_{MOTOR}	200 Vollschritte/Umdrehung

$$\begin{aligned}
 n &= \frac{f_{VS} \cdot 60}{S_{MOTOR}} \\
 &= \frac{1800 \text{ Hz} \cdot 60}{200} = 540 \text{ min}^{-1}
 \end{aligned}$$

Diese Drehzahl gilt für den Halbschrittbetrieb. Im Mikroschrittbetrieb sinkt das Drehmoment bei höherer Drehzahl weiter ab [15]. Beim Test des Motors wurde ermittelt, dass dieser bei einer Drehzahl von

$$n = 640 \text{ min}^{-1}$$

ein ausreichendes Drehmoment zur Bewegung des Z-Trägers aufbringt.

6.1.5 Webcam

Um die grafische Definition des Messbereichs zu vereinfachen, wurde der Anlage eine Kamera hinzugefügt, mit der der Benutzer ein Foto des Testaufbaus im unteren Bereich der Anlage aufnehmen kann. Die Kamera ist dabei an einer speziell angefertigten Vorrichtung angebracht, die am Z-Rahmen befestigt werden kann (s. Abbildung 26).

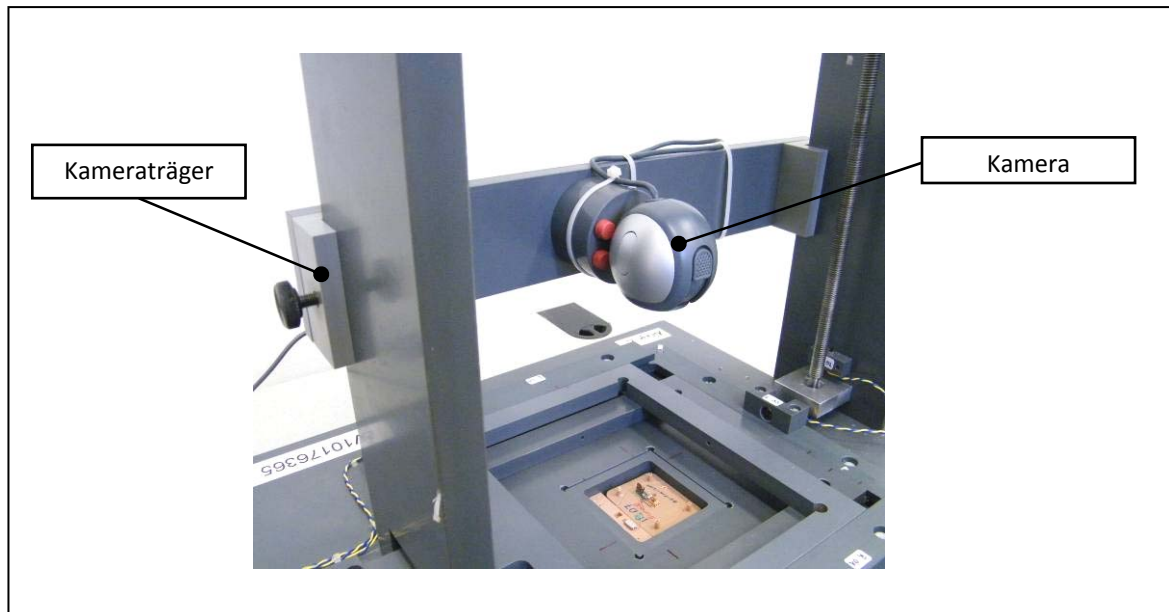
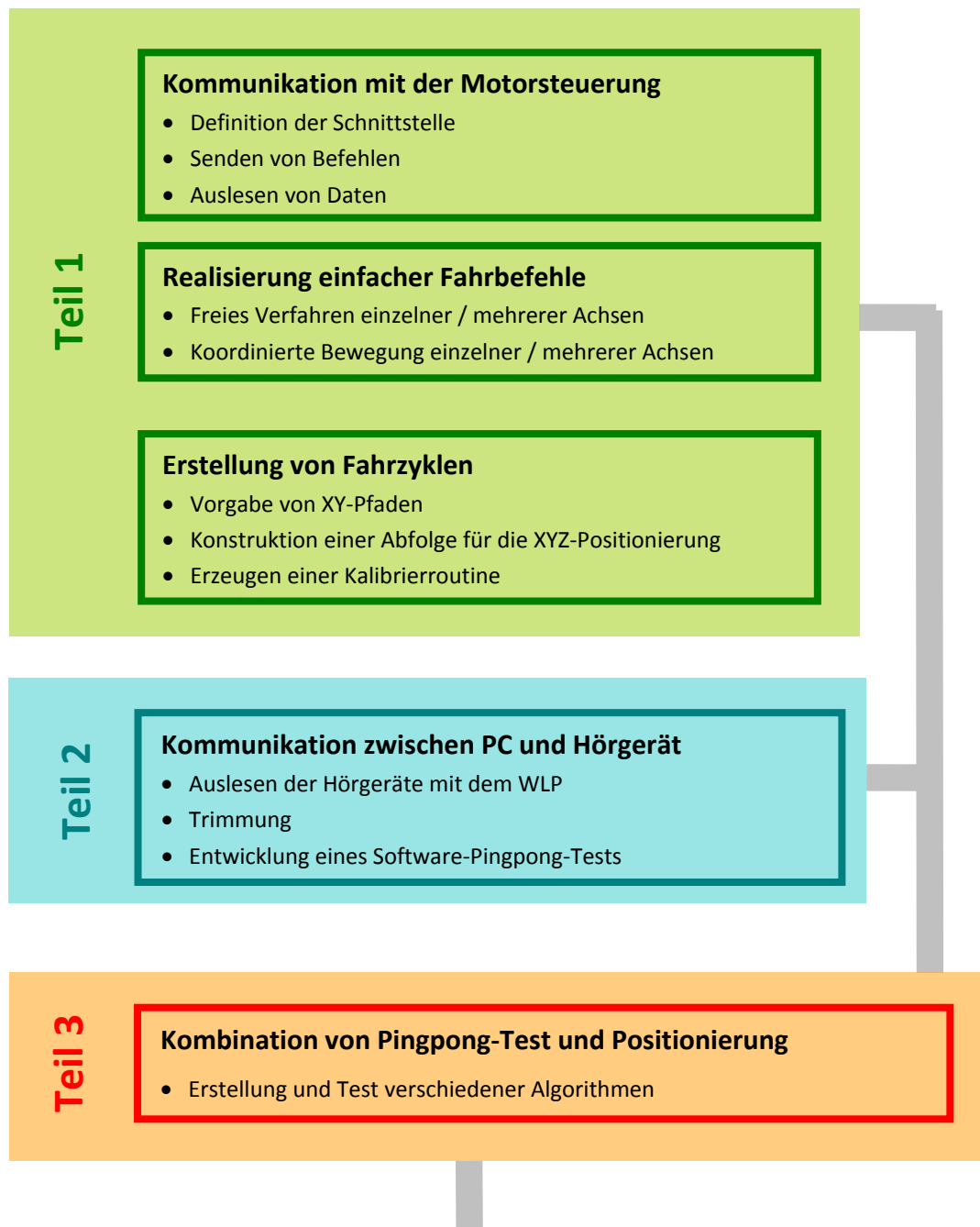


Abbildung 26: Webcam und -träger

6.2 Software

Grundvoraussetzung für die Automatisierung des Messablaufes ist die Erstellung eines aufgabenorientierten Programmkonzepts. Als Grundlage für die Erstellung der Anlagensoftware wird dabei die Programmierplattform MATLAB verwendet, da diese in Bezug auf die Anforderungen eine Vielzahl geeigneter Funktionen und Hilfsmittel enthält. Die Umsetzung des Programmkonzeptes erfolgte in Zusammenarbeit mit den späteren Hauptanwendern der Anlage.

Für die schrittweise Umsetzung der Automatisierung kann das Gesamtkonzept in folgende Abschnitte unterteilt werden (s. nächste Seite):



(Fortsetzung nächste Seite)

(Fortsetzung)

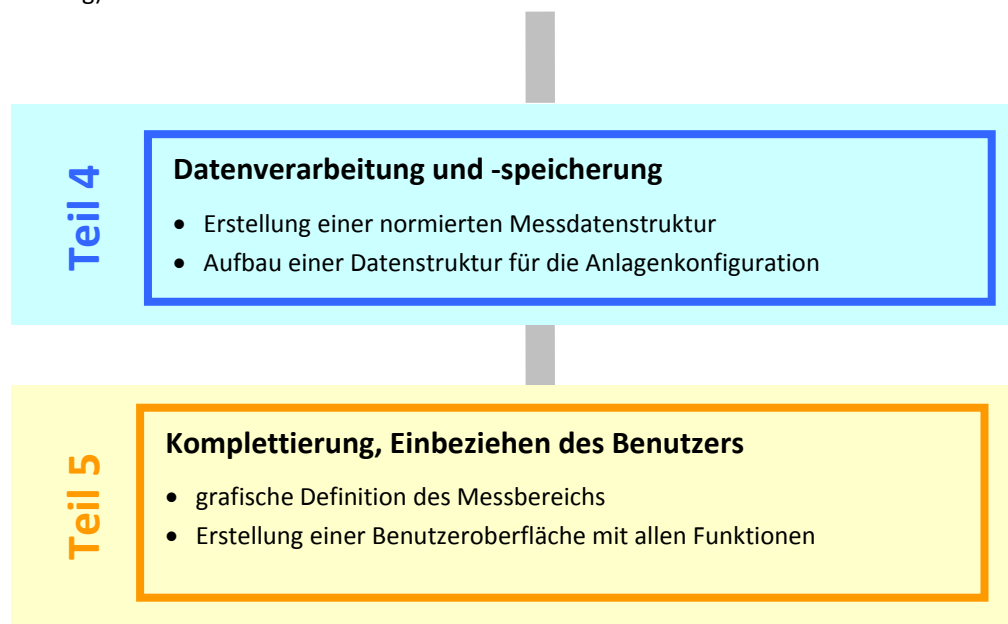


Abbildung 27: Softwarekonzept

6.2.1 Kommunikation mit der Motorsteuerung

6.2.1.1 Schnittstellendefinition

Die Verbindung zwischen Mess-PC und der Schrittmotor-Steuerkarte erfolgt über die RS232-Schnittstelle. Hierbei muss am PC der entsprechende COM-Port ausgewählt und konfiguriert werden, da je nach Anschlussprinzip (RS232-Buchse, USB-Adapter, ...) die Portnummer variieren kann. Auf MATLAB-Ebene gibt es keine Möglichkeit die Portnummer des angeschlossenen Moduls automatisch zu ermitteln, daher muss dies bei jedem Start der Software vom Benutzer vorgenommen werden. Als optimale Übertragungsrate sind 115200 Baud eingestellt.

6.2.1.2 Senden von Maschinenbefehlen

Für die Positionierung der Achsen ist idealerweise eine zyklische Datenübertragung (Polling) notwendig. Speziell beim Motorlauf sollte die aktuelle Position permanent abgefragt werden, um feststellen zu können ob der Motor seine Zielposition erreicht hat. Da Echtzeit-Polling am PC nicht möglich ist, konnte dies für die Anlage jedoch nicht realisiert werden.

Bei Schnittstellentests kam es in den meisten Fällen zu Timeout-Fehlern während der Leseoperationen, die sich jedoch nach Rücksetzen der Schnittstelle und erneutem Senden nicht wiederholten.

Um daher eine annähernd zeitnahe und fehlerfreie Kommunikation zu gewährleisten, wurde die Basisfunktion `send` erstellt, mit der im Rahmen der neuen Software sämtliche Lese- und Schreiboperationen der Schnittstelle durchgeführt werden.

Die `send`-Funktion beinhaltet die gängigsten Befehle der TMCL-Programmiersprache (Vgl. [5]). Dies hat zum Zweck, die aufwendige Generierung der Befehlsbytefolge (s. 3.3.3) sowie die Umrechnung des Rückgabewertes zu ersparen. Gleichzeitig bietet sie mit vorgefertigten Syntaxen eine einfachere Ansprechbarkeit durch den Entwickler.

Im Falle eines Übertragungsfehlers wiederholt die Funktion den Vorgang bis zu fünfmal bis eine Fehlermeldung angezeigt und das Programm abgebrochen wird (s. Abbildung 28). Diese Schleife erwies sich im Zuge der Entwicklung als sinnvoll und stabilisierend bezüglich zyklischer Datenabfragen.

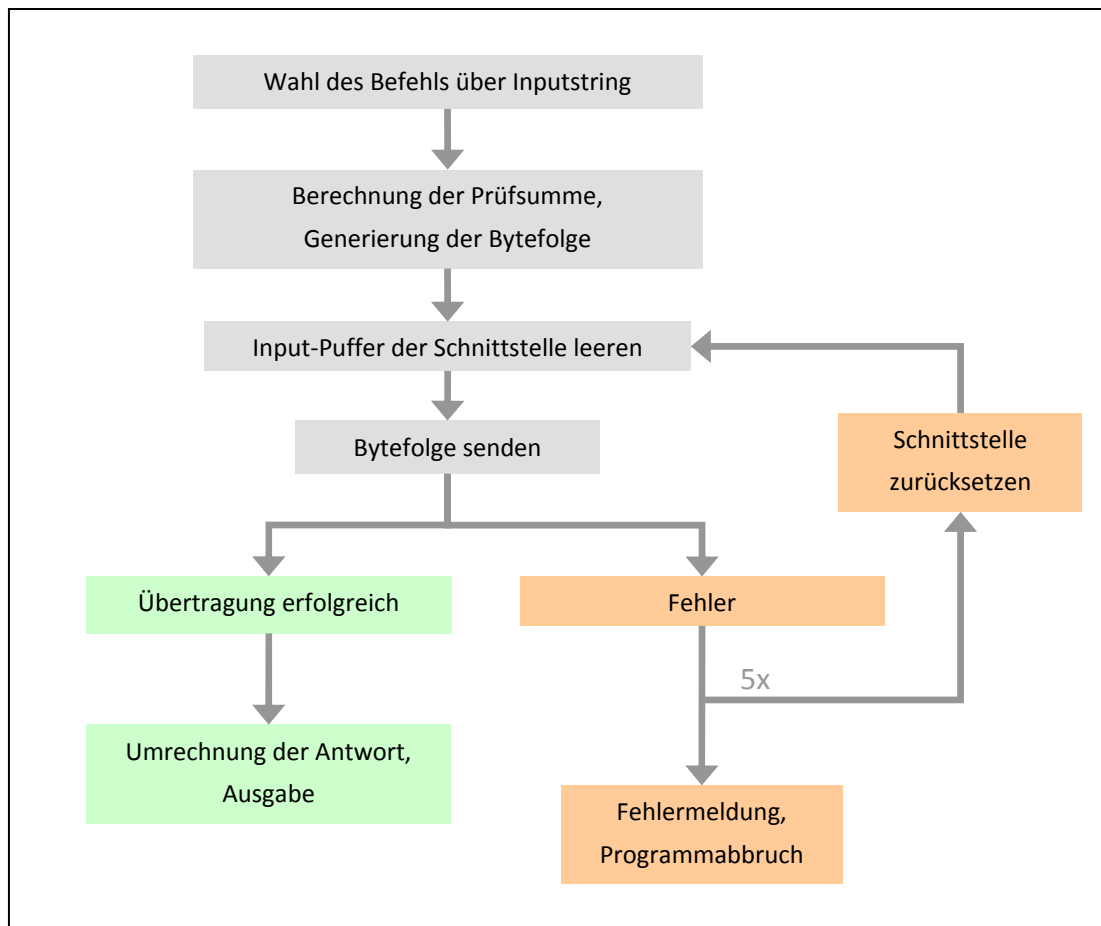


Abbildung 28: Ablaufschema der send-Funktion

6.2.2 Realisierung einfacher Fahrbefehle

6.2.2.1 Freies Verfahren der Achsen

Zielstellung hierbei ist es, dem Benutzer eine einfache Möglichkeit zu bieten, Messsonde (X- und Y-Achse) sowie Referenzgerät (Z-Achse) frei bewegen zu können. Diese Funktionalität soll später auch in die GUI aufgenommen werden, daher wurden zwei Methoden entwickelt, die Motoren positionsunabhängig zu verfahren. Gesteuert werden kann demnach

- über die Pfeiltasten der PC-Tastatur und
- über in die GUI integrierte Schaltflächen.

Der TMCL-Befehlssatz enthält dafür zwei Bewegungskommandos um die Motoren in positive oder negative Richtung zu rotieren bzw. zu stoppen (*ROR*, *ROL*, *MST*). Entsprechend der Festlegung in 6.2.1.2 werden diese mit der Funktion `send` an das TMCM-Steuermodule gesendet.

Die Tastensteuerung wird in MATLAB über die Funktion `KeyPressFcn` bzw. `KeyReleaseFcn` umgesetzt (Vgl. [16]). Diese ermöglichen das Ausführen von Benutzerfunktionen beim Drücken und Loslassen bestimmter Tasten.

6.2.2.2 Koordinierte Bewegung der Achsen

Die Motorbewegung soll hierbei unter Vorgabe einer Zielposition erfolgen. Da sich die Z-Achse hinsichtlich Fahrbereich und Geschwindigkeit von X- und Y-Achse unterscheidet, wurden zwei Funktionen erstellt:

Die Funktion `go2coord` dient zur Positionierung von X- und Y-Achse. In Bezug auf den späteren Messbereich wird diese zum Abfahren der einzelnen XY-Messpunkte verwendet. Beide Achsen werden dabei gleichzeitig bewegt. Die Geschwindigkeit der beiden Motoren wird von der Steuerung selbst gewählt, sodass die Motoren ihre Zielposition zur gleichen Zeit erreichen. Basis dafür sind zwei in der TMCL integrierte Kommandos (*SCO*, *GCO*, *MVP*).

Die Funktion `go2z` verfährt nur die Z-Achse und wird im Messablauf daher für die Positionierung des Referenzgerätes benutzt. Sie weist die gleiche Struktur wie `go2coord` auf, verwendet auf TMCL-Ebene jedoch nur den Befehl *MVP*, der den Motor an eine vorgegebene Position bewegt.

Beide Funktionen verfügen zusätzlich über die Option, die Motoren relativ zu positionieren. Dies muss beim Aufruf entsprechend angegeben werden.

6.2.3 Erstellung von Fahrzyklen

6.2.3.1 Vorgabe von XY-Pfaden

Mit Hilfe der Funktion `go2coord` können im späteren Messablauf die jeweiligen XY-Punkte des Messrasters angefahren werden. Grundlage hierfür ist eine Datenstruktur, die sich zusammensetzt aus:

- den Messpunkten, an denen der Pingpong-Test durchgeführt wird und
- Pfadpunkten, die jeweils nur überfahren werden um an weiter entfernte Messpunkte zu gelangen

Beide Koordinatenpaarungen sind in einer einheitlichen Datenstruktur gespeichert, die im Automatikbetrieb schleifenweise abgearbeitet werden.

6.2.3.2 Konstruktion einer Abfolge für die XYZ-Positionierung

Kern der Messroutine ist die Funktion `automatedMode`. Diese enthält die schleifenweise Abarbeitung des Messrasters und wird im weiteren Verlauf der Softwareentwicklung um die nötigen Funktionen (Pingpong-Test, Datensicherung, etc.) erweitert.

Folgende Punkte sind in dieser Funktion von besonderer Bedeutung:

„Warten“ auf den Motor

Um im Programmablauf festzustellen, ob der Motor seine Zielposition erreicht hat, muss dies möglichst permanent abgefragt werden. Die asynchrone Datenübertragung mit der TCM-Steuerkarte war nicht realisierbar daher wird nach senden des Fahrbefehls ein sog. „Position-erreicht-Flag“ anhand einer `while`-Schleife abgefragt. Dieses Flag wird von der Motorsteuerung nach Erreichen einer vorgegebenen Zielposition gesetzt und muss mit der `send`-Funktion entsprechend abgefragt werden. Zusätzlich zu diesem Flag wird in der Schleife eine weitere Variable abgefragt, die in der späteren GUI nach Drücken eines Stopp-Buttons gesetzt wird. Zweck dieser Schleife ist, den Programmablauf zu stoppen solange der Motor noch fährt.

Anhalten des Ablaufs

Um den Ablauf jederzeit fehlerfrei abbrechen zu können, wurde im MATLAB-Root-Objekt eine dafür vorgesehene Variable erstellt. Das Root-Objekt ist ein dauerhaft vorhandenes *Structure Array*, das beim Start von MATLAB erzeugt wird und hauptsächlich Grundeinstellungen enthält. Im Gegensatz zu global definierten Variablen sind die Variablen im Root-Objekt nicht abhängig von der aktuell laufenden Funktion, d.h. nicht Workspace-abhängig. Änderungen an diesen Variablen werden sofort aktualisiert und können von jeder Funktion abgerufen werden.

Im Messablauf wird die erstellte Root-Variable (Stopp-Flag) in allen beteiligten Funktionen und Unterfunktionen regelmäßig mit dem Befehl `getappdata` abgefragt. Zum Abbrechen des Ablaufes wird das Stopp-Flag mit `setappdata` gesetzt. Unabhängig von der aktuell laufenden Funktion wird, bedingt durch den Flag-Status, bei der nächsten Abfrage die Routine zum Abbruch (Motorstopp, ...) gestartet.

Pausieren und Fortsetzen

An jedem Messpunkt, d.h. bei jedem Schleifendurchlauf der XY-Positionierung wird zusammen mit den XYZ-Werten eine Zählervariable gespeichert, die die Anzahl der bereits erfolgten Schleifendurchläufe enthält. Anhand dieser Variablen kann der Messablauf an

jeder beliebigen Stelle angehalten und fortgesetzt werden. Dies ist besonders bei Abläufen mit vielen Messpunkten von Vorteil.

6.2.3.3 Erzeugung einer Kalibrieroutine

Die Kalibrierung der Achsen ist notwendig um deren Fahrbereich zu definieren. Hierbei werden die manuell ausgemessenen, festen Dimensionen der Achsen (mm) mit der Gesamtschrittzahl abgeglichen. Generell ist die Kalibrierung nach jedem Einschaltvorgang erforderlich und erfolgt mittels einer Referenzfahrt (s. Abbildung 29).



Abbildung 29: Ablaufschema der Kalibrieroutine

Der Skalenabgleich beinhaltet die Berechnung eines Umrechnungsfaktors cal zwischen Millimetern und Schritten nach dem folgenden Ablauf:

$$cal_x = \frac{x_{MAX} - x_{MIN}}{x_{DIM}}$$

x_{MAX} Maximalwert der Achse [Schritte]

x_{MIN} Minimalwert der Achse [Schritte]

x_{DIM} Gemessene Achsenlänge [mm]

Die Berechnung der Faktoren cal_y für die Y-Achse und cal_z für die Z-Achse erfolgt analog.

Als Grundeinheit der Software wurde Millimeter gewählt. Positionswerte die an die Steuerkarte gesendet bzw. von der Steuerkarte empfangen werden, müssen daher vor jedem Sende- bzw. Anzeigevorgang umgerechnet werden. Um eine allgemeingültige Funktion zur Umrechnung der Längeneinheiten zu schaffen, wurde die Funktion `convert` erstellt. Diese ermöglicht die gegenseitige Umrechnung zwischen

- Millimetern, da dies als Grundeinheit des Systems gewählt wurde,
- Zentimetern,
- Schritten, da dies die Grundeinheit der Positionierbefehle an die Steuerkarte ist, und
- Pixel für die Anzeige der Positionen im Kamerabild.

6.2.4 Kommunikation zwischen PC und Hörgerät

Um den Pingpong-Test per Software steuer- und auswertbar zu machen, wird mit Hilfe des WLP eine Statusabfrage von Referenzgerät und Messsonde durchgeführt. Das Referenzgerät ist dabei grundsätzlich als rechtes Hörgerät definiert, die Messsonde als linkes. Für die Kommunikation mit den Geräten existiert auf MATLAB-Ebene eine dafür konzipierte, SAT-interne Funktion namens `send_frame`. Diese liefert als Output eine Bytefolge zurück, die Informationen des rechten und linken Gerätes als Nutzdaten („Payload“) enthält. Ausgewertet wird Byte 1 und 6 des Payloads. Diese sind bei Erkennung beider Geräte gleich (s. Tabelle 14).

Tabelle 14: Wert von Byte 1 und 6 des `send_frame`-Payloads

Fall:	Messsonde in Reichweite	Messsonde außer Reichweite	Referenzgerät außer Reichweite des WLP
Byte 1 und 7	[07... 07 ...]	[07... 00 ...]	[00 ... 00 ...]

6.2.5 Automatisches Trimmen

Das Trimmen von Referenzgerät und Messsonde erfolgt analog zur Statusabfrage mit Hilfe der `send_frame`-Funktion und kann so auf einfache Weise in den Programmablauf integriert werden [17].

6.2.6 Kombination von Pingpong-Test und Positionierung

Für den Pingpong-Test und die davon abhängige Positionierung der Z-Achse wurde die Funktion `bt_measure` erstellt. Diese wird an jeder XY-Position (innerhalb der `automatedMode`-Funktion) aufgerufen um den Test durchzuführen und die Z-Achse entsprechend zu positionieren. Der Gesamtablauf der automatischen Messung gestaltet sich nach dem Schema in Abbildung 30.

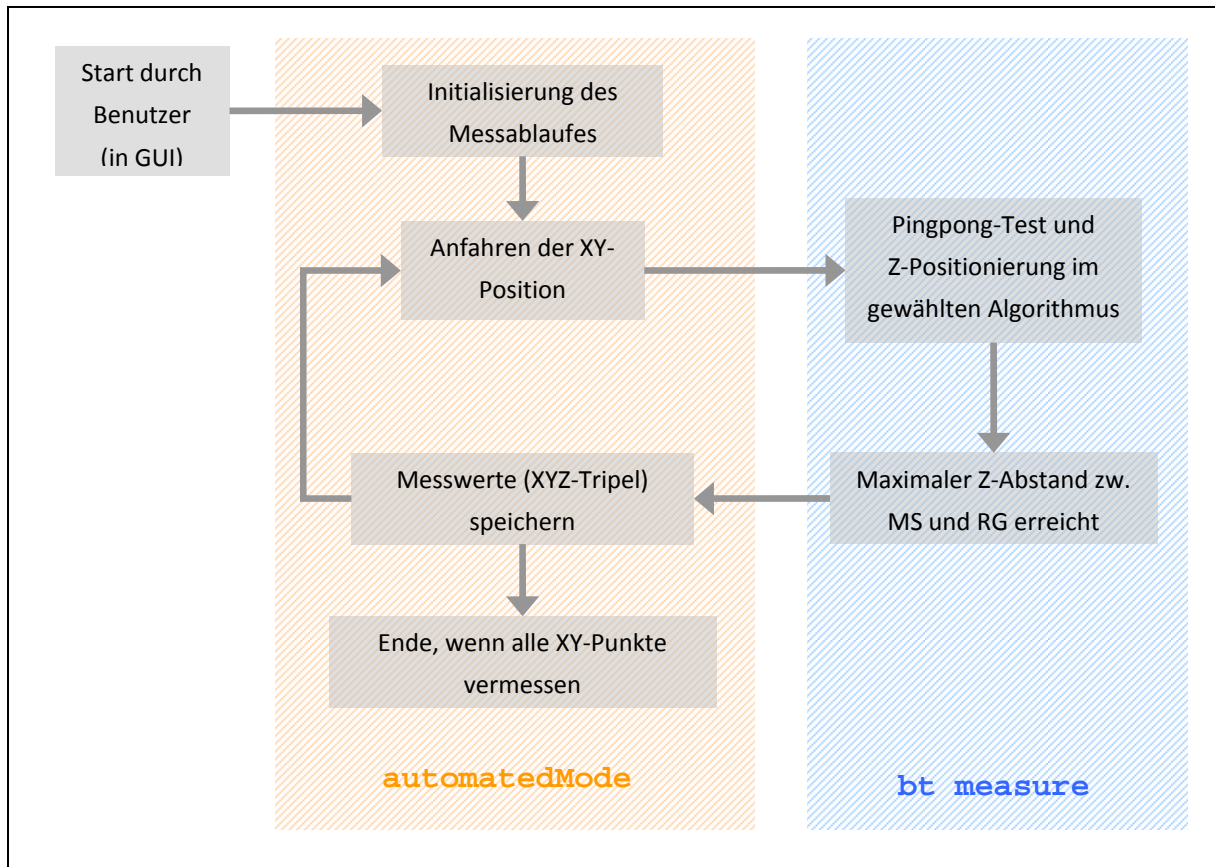


Abbildung 30: Ablaufschema des automatischen Messablaufs

Die `bt_measure`-Funktion enthält verschiedene Algorithmen für die testabhängige Positionierung der Z-Achse. Erstellt wurden die Algorithmen nach den folgenden Vorgabekriterien:

- Messwertesicherheit und –kontinuität (Algorithmus 1)
- geringe Messzeit (Algorithmus 2) und
- Flexibilität hinsichtlich des Ablaufes (Algorithmus 3).

Die Auswahl des Algorithmus trifft der Benutzer vor dem Start der Messung bzw. kann die Nummer des Algorithmus in den Einstellungen verändert werden. Abbildung 31 zeigt die unterschiedlichen Ablaufpläne der Algorithmen. Tabelle 15 enthält eine Beschreibung der Ereignisse in den jeweiligen Algorithmen.

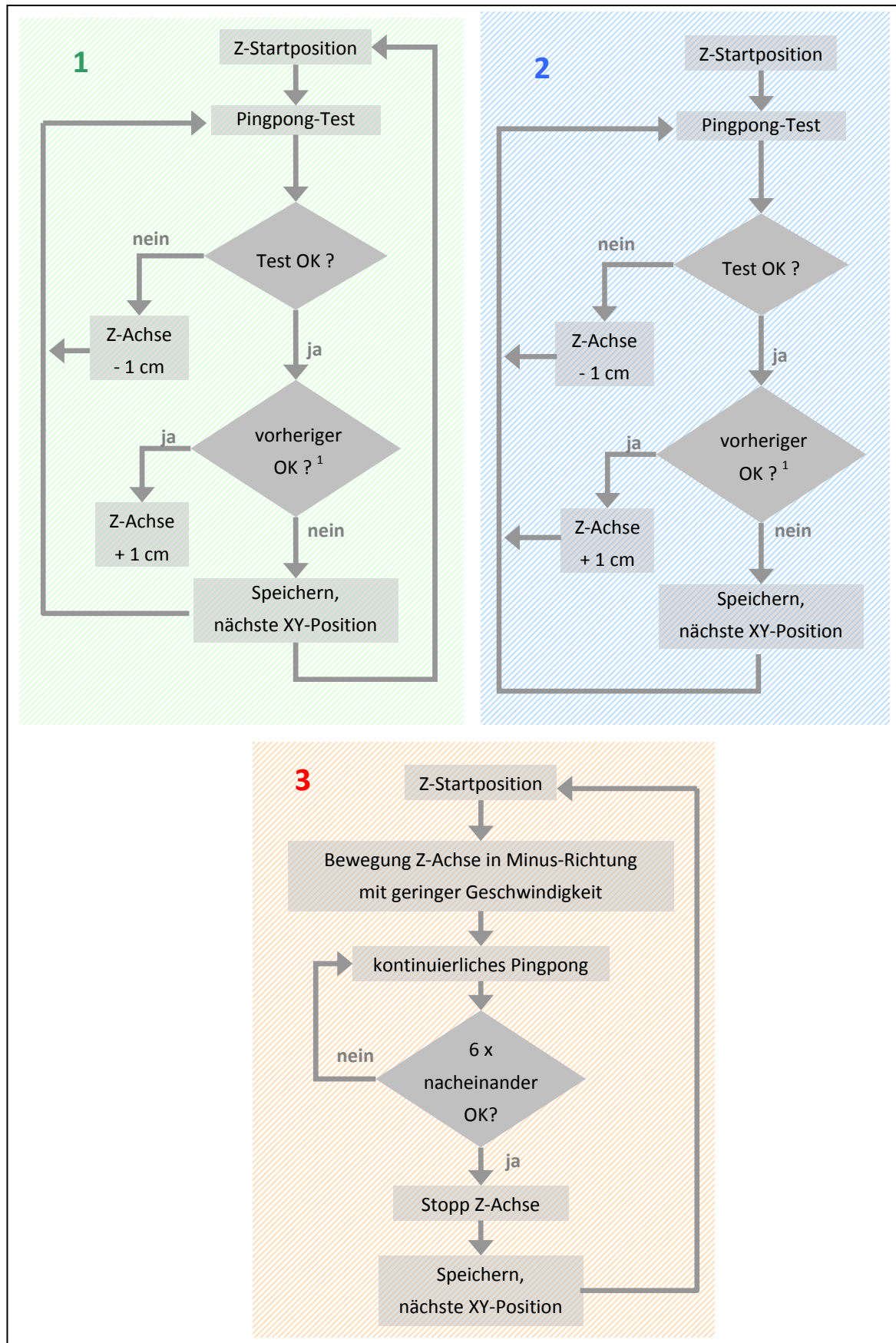


Abbildung 31: Ablaufschema Algorithmus 1, 2 und 3

Zu ¹ : Beim ersten Durchlauf (kein vorheriger Test) gilt: OK: Z-Achse + 1cm; NOK: Z-Achse - 1cm

Tabelle 15: Ereignisse der Algorithmen

	Algorithmus 1	Algorithmus 2	Algorithmus 3
Prinzip	Schrittweise Annäherung RG → MS	Schrittweise Annäherung RG → MS	Kontinuierliche Annäherung RG → MS mit permanentem Pingpong
Z-Startpos.	gleiche Startposition bei jedem Test	Position des vorherigen Tests	gleiche Startposition bei jedem Test
Abbruch wenn	OK + vorheriger NOK	OK + vorheriger NOK	6 x hintereinander OK
Z+ wenn	OK,	OK,	-
	OK + vorheriger OK,	OK + vorheriger OK,	
Z- wenn	NOK,	NOK,	-
	NOK + vorheriger NOK,	NOK + vorheriger NOK,	
	NOK + vorheriger OK	NOK + vorheriger OK	

6.2.7 Datenverarbeitung und Speicherung

6.2.7.1 Erstellung einer normierten Messdatenstruktur

Um die gemessenen Werte in einfacher Weise auswerten zu können und gleichzeitig Informationen über die Messung zu erhalten, wurde eine Datenstruktur aufgebaut, die für jede Messung vereinheitlicht ist. Dieses Datenobjekt hat in MATLAB die Form eines *Structure Arrays* und enthält u. A. die folgenden Informationen:

- Messpunkte und Pfadpunkte, die vom Benutzer vorgegeben werden
- die gemessenen XYZ-Wertetripel
- Fortschritt einer bereits begonnenen Messung
- Informationen über das Testobjekt (Bauteiltyp, -seriennummer, ...)
- Datum, Startzeit, Dauer der Messung
- Kamerabild (als Bildmatrix)

Zu Beginn jeder Messung wird die Datenstruktur durch die Funktion `generateData` erzeugt und danach als Input- bzw. Outputvariable (unter der generellen Bezeichnung `data`) zwischen den entsprechenden Funktionen übergeben. Zum Speichern der Struktur wurde die Funktion `exportData` erstellt, die die Daten als:

- MATLAB-Datei (*.mat),
- kommaseparierte Liste (*.csv),
- XML-Datei (*.xml),
- MS Word-Dokument (*.doc) und
- HTML-Datei (*.html)

speichern kann.

6.2.7.2 Aufbau einer Konfigurationsstruktur

Um die Einstellungen der Anlage zu sichern, wurde, ähnlich zur Messdatenstruktur, ein Datenobjekt zusammengestellt, das alle Konfigurationswerte enthält. Dieses wird in den Funktionsaufrufen als In- bzw. Output übergeben. Somit werden Veränderungen an den Einstellungen schnell erfasst und Momentanwerte funktionsübergreifend bekannt gemacht. Die enthaltenen Einstellungen umfassen:

- Systemgrenzen (festgelegte, maximale Fahrbereiche der Achsen)
- Kalibrierwerte (Werte, die bei der Referenzfahrt erfasst werden)
- Schnittstellenparameter
- Motorparameter
- Messeinstellungen (Anzahl der Pingpongs an jeder Z-Position, Z-Schrittweite, Intervall für Trimmung, ...)
- Webcam-Einstellungen (für Umrechnung von Pixel in Millimeter)
- aktuelle Positionswerte
- Flags (zur Anzeige laufender Messungen oder zur Bestätigung einer durchgeführten Kalibrierung)

Die Konfigurationsstruktur ist in einer Initialisierungsdatei (*.ini) gespeichert und wird beim Start der Anlagensoftware mit der Funktion `readConfig` eingelesen. In allen Funktionen, die die Konfigurationsdaten benötigen, wird die Struktur als *Structure Array* mit dem einheitlichen Namen `hades` (abgeleitet aus der Anlagenbezeichnung: „Hearing Aid Distance Evaluation System“) geführt.

Da sich die beiden Anlagen hinsichtlich der Z-Achse unterscheiden, existieren dementsprechend auch zwei verschiedene Initialisierungsdateien (`config1.ini`, `config2.ini`). Vor dem Laden der Datei wird daher mittels einer in der TCM-Steuerkarte gespeicherten Nummer die Anlagen-ID abgefragt.

Werden Einstellungen verändert, wird die Konfigurationsstruktur mit der `writeConfig`-Funktion wieder in der Initialisierungsdatei abgespeichert.

6.2.8 Komplettierung, Einbeziehen des Benutzers

6.2.8.1 Grafische Definition des Messbereichs

Um die Definition des Messbereiches am Testobjekt zu erleichtern, wird vor jeder Messung ein Bild des Testaufbaus mit der Funktion `takeImage` aufgenommen. Zum Auslesen der Kamera verwendet diese Funktion einen in der Programmiersprache *Python* geschriebenen Algorithmus. Da das aufgenommene Bild nicht der Größe des Messbereiches entspricht (s. Abbildung 32), muss dieses entsprechend skaliert werden. Zu diesem Zweck sind auf der XY-Ebene der Anlage zwei feste Markierungen angezeichnet, die die Punkte $(0; 0)$ und $(X_{\max}; Y_{\max})$ angeben (s. Abbildung 33). Nach der Bildaufnahme wählt der Nutzer die Markierungen im Bild per Mausklick an, wodurch das Bild anschließend skaliert wird.

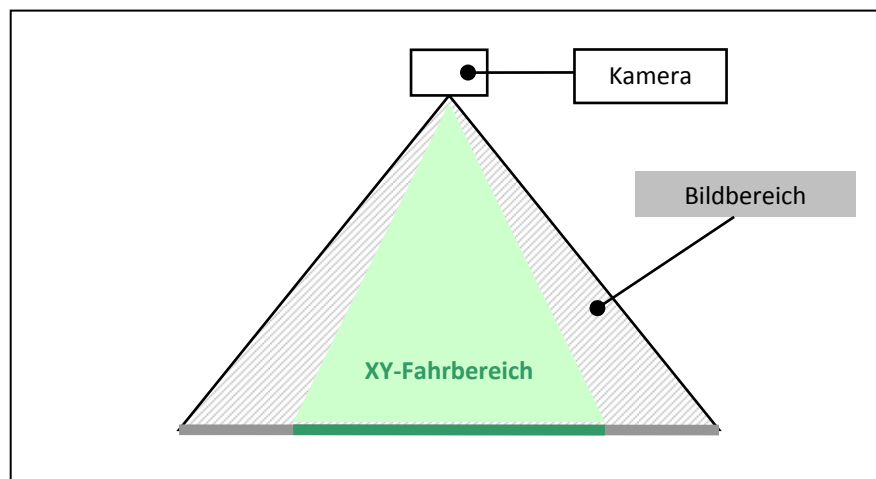


Abbildung 32: Übergröße des Kamerabildes

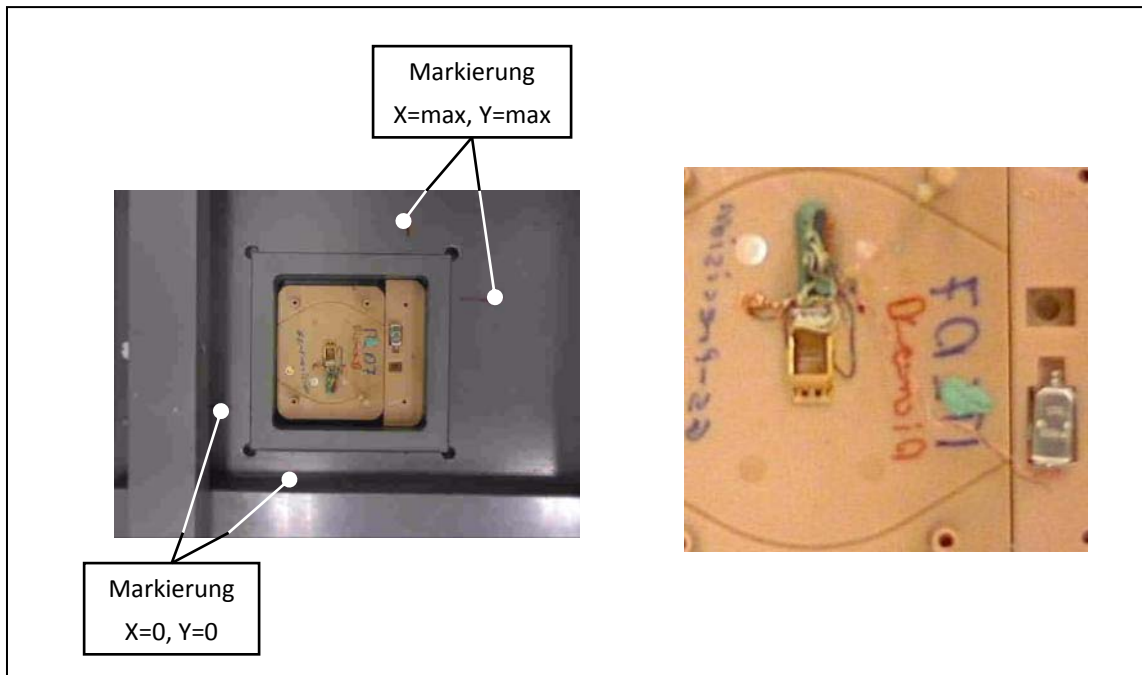


Abbildung 33: Kamerabild unskaliert (l.), skaliert (r.)

Nach der Skalierung kann der Nutzer den Messbereich über drei verschiedene Methoden definieren:

Methode 1: Eingrenzen einer Fläche

Hierbei wird im Kamerabild über mindestens 3 Punkte eine Fläche festgelegt, aus der anschließend die Messpunkte berechnet werden. Möglich ist hierbei auch die Erzeugung eines Pfades über nur 2 Punkte.

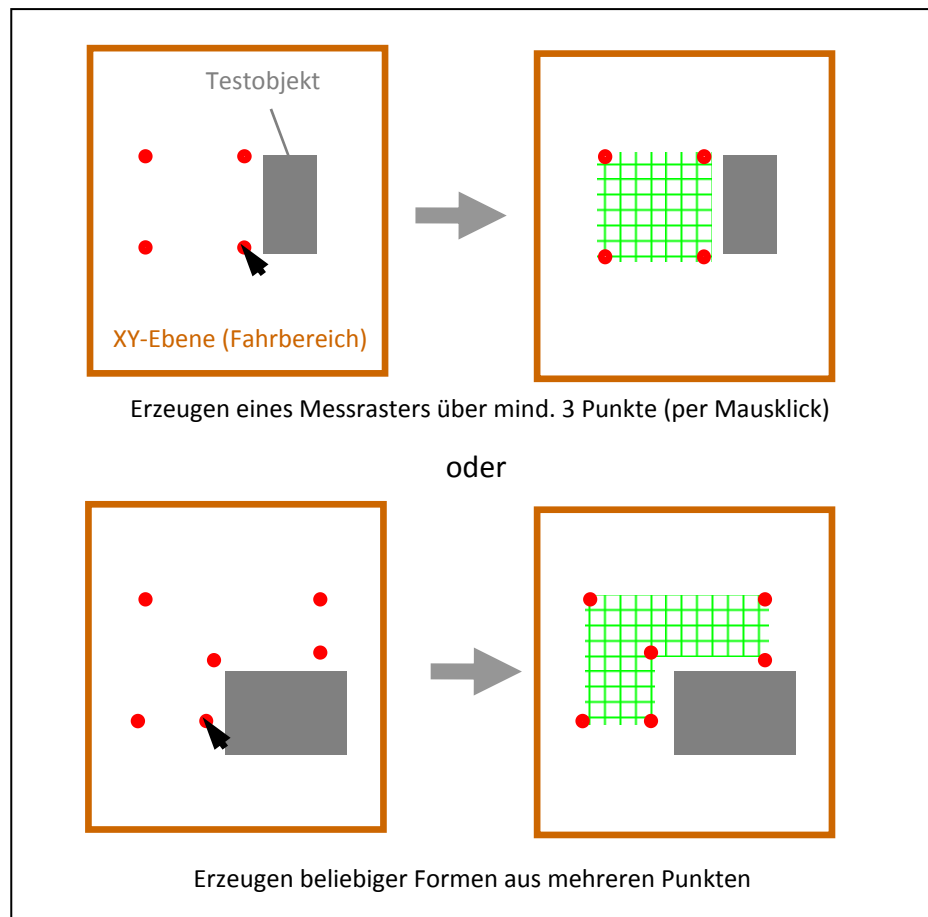


Abbildung 34: Methode 1

Die Definition des Messrasters erfolgt schrittweise in einer externen GUI (s. Abbildung 35). Zunächst werden dort die Flächenpunkte per Mausklick markiert. Die Funktion `findWaypoints` erzeugt aus dieser Fläche das Messraster dessen Genauigkeit, d.h. der Abstand zwischen zwei Messpunkten eingestellt werden kann. Des Weiteren generiert die `findWaypoints`-Funktion den abzufahrenden Pfad.

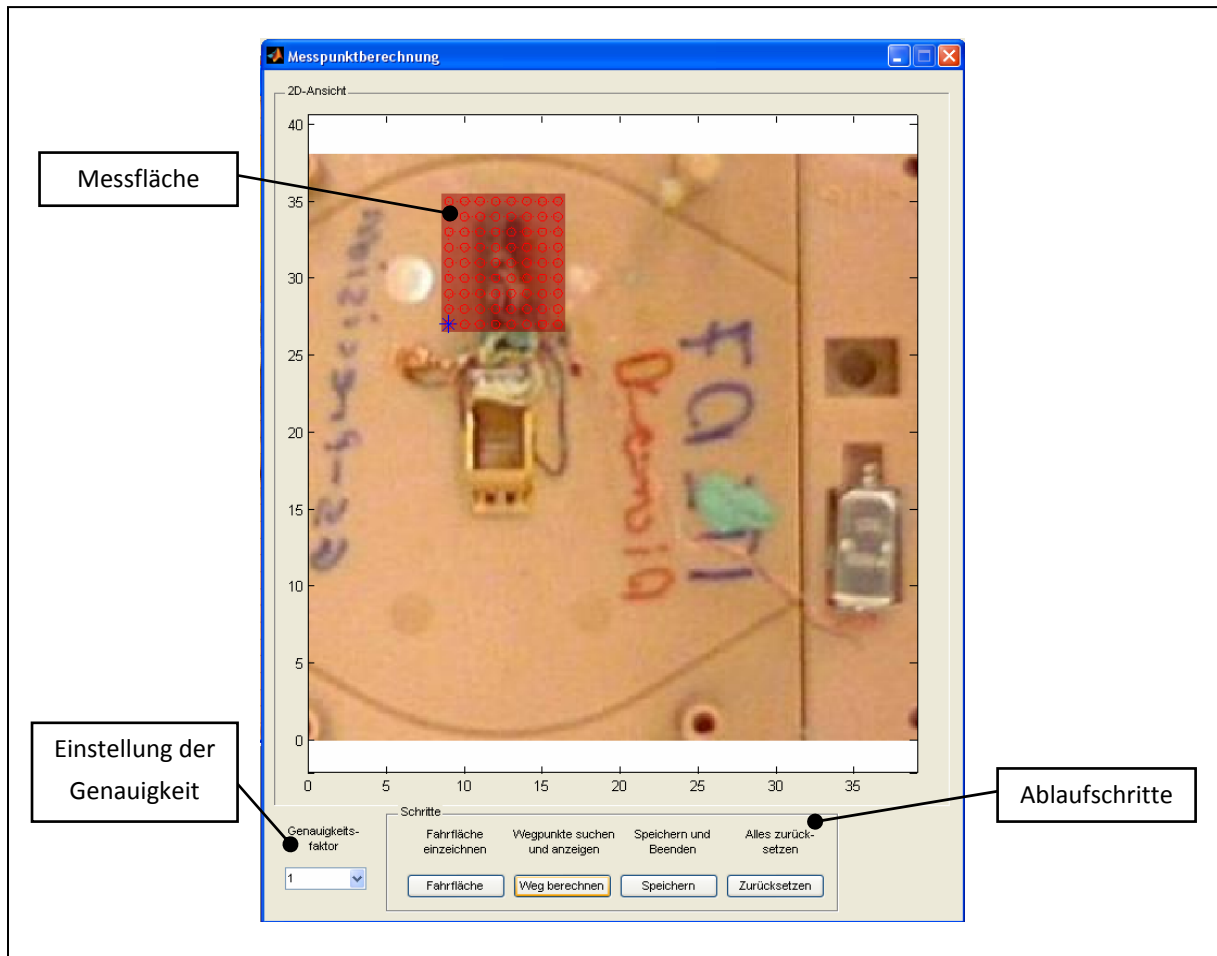


Abbildung 35: GUI zur Messrasterdefinition nach Methode 1

Methode 2: Festlegen einzelner Messpunkte

Der Benutzer legt bei dieser Methode die Mess- und Pfadpunkte einzeln in einem Millimeterraster fest.

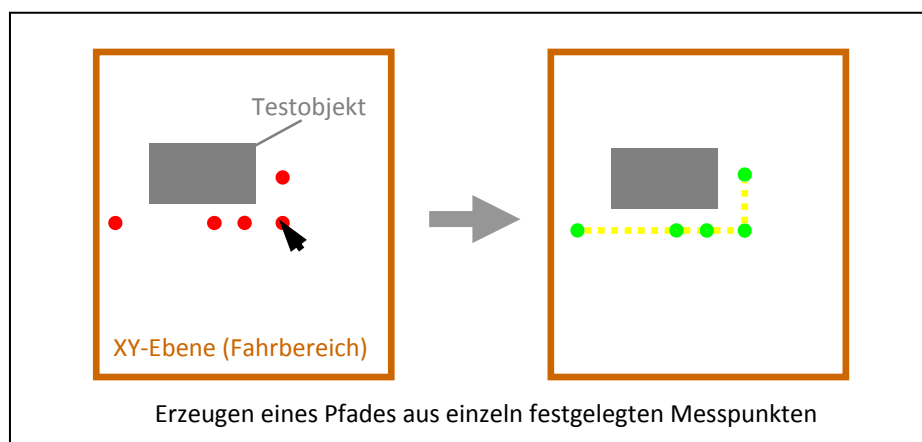


Abbildung 36: Methode 2

Die einzelnen Punkte werden hierbei in einem externen Fenster (Standard Figure-Fenster mit dem Kamerabild) per Mausklick markiert. Die Funktion `recordPath` zeichnet dabei die Koordinaten der Mausklicks auf und speichert diese in der Messdatenstruktur als Weg- bzw. Pfadpunkte.)

Methode 3: Anfahren eines Hindernisbereiches

Bei dieser Methode kann der Benutzer den Messbereich um das Testobjekt durch die Positionierung der Messsonde definieren. Die angefahrene Position wird dabei als Grenzpunkt des Hindernisbereiches gespeichert. Nachdem mindestens 4 Punkte definiert wurden, erzeugt der Algorithmus das Messraster *um* den eingegrenzten Bereich. Die Breite des Bereiches ist einstellbar (Dropdown-Menü in GUI).

Hierbei können die angefahrenen Punkte als Grenzpunkte des Hindernisbereiches erfasst werden.

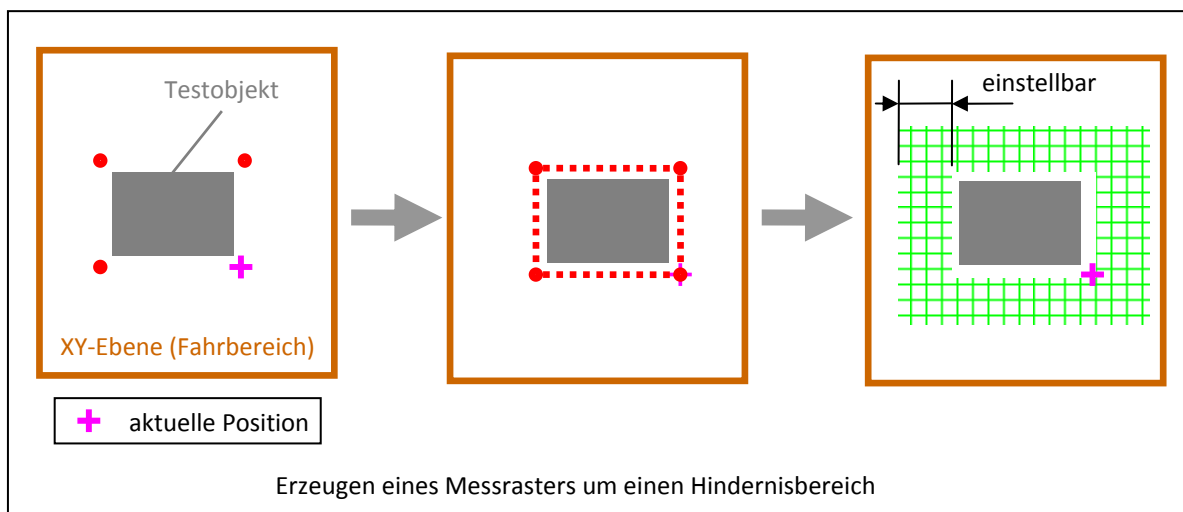


Abbildung 37: Methode 3

Realisiert wird die Messrastererstellung durch die `drawObstacles`-Funktion. Diese erzeugt zunächst ein Raster über die gesamte Bildfläche. Aus dieser wird anschließend der Hindernisbereich ausgeschlossen.

Die Mess- und Pfadpunkte werden anschließend in der Messdatenstruktur (Vgl. 6.2.7.1) gespeichert.

6.2.8.2 Erstellung der Benutzeroberfläche

Die neue Benutzeroberfläche wurde mit dem, in MATLAB integrierten Assistenten GUIDE erstellt. Dieser erzeugt automatisch die Subfunktionen zur Steuerung der verschiedenen Komponenten und legt die sogenannte handles-Struktur an. Dieses *Structure Array* enthält alle Daten der Steuerelemente sowie die Konfigurations- und die Messdatenstruktur (s. 6.2.7.1 bzw. 6.2.7.2). Dies ermöglicht einen einfacheren Zugriff auf die Daten, da die handles-Struktur jeder Subfunktion übergeben wird.

Die GUI baut sich wie folgt auf:

I Hauptfenster

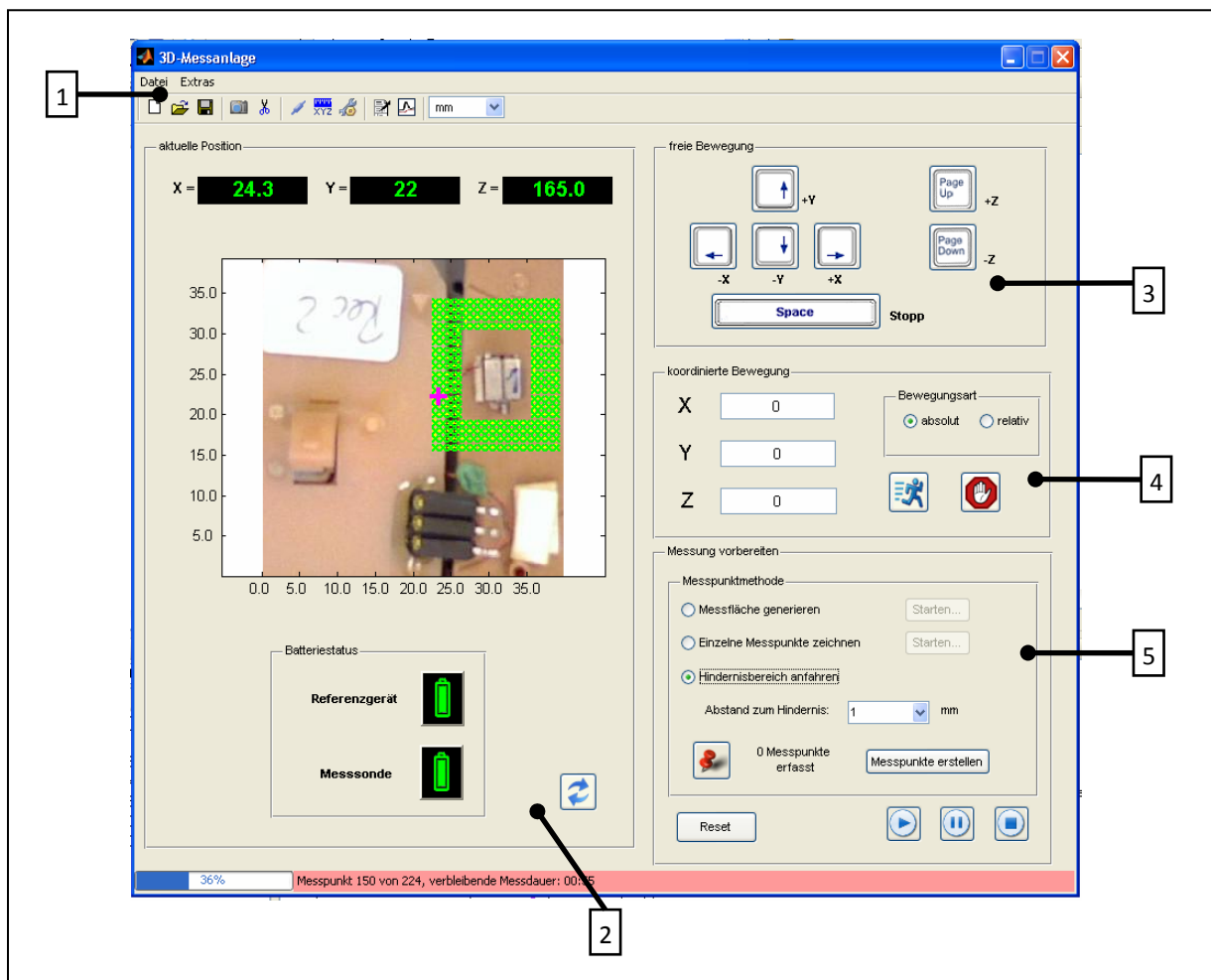


Abbildung 38: Hauptfenster PC-Benutzeroberfläche

Tabelle 16: Bereiche der GUI

Nr.	Bezeichnung
1	Menüleiste und Toolbar
2	Anzeigebereich
3	Freies Verfahren
4	Koordiniertes Verfahren
5	Messungen

(1) Menüleiste und Toolbar:

Tabelle 17: Komponenten der Toolbar

	Unterpunkte: Neu, Öffnen, Speichern, Speichern unter..., Beenden
	Unterpunkte: Header bearbeiten, Auswertesoftware starten, Einstellungen bearbeiten
	Standardfunktionen zum Erstellen / Öffnen / Speichern der Messdaten (Messdatenstruktur)
	Webcambild aufnehmen
	Webcambild auf den XY-Fahrbereich skalieren
	Verbindung zur TMCM-Modul herstellen / trennen
	Kalibrierung (Referenzfahrt starten)
	Einstellungen öffnen
	Dateiheader bearbeiten
	Auswertesoftware starten
	Auswahl der Einheit für koordinierte Positionierung und die Positionsanzeige

(2) Anzeigebereich:

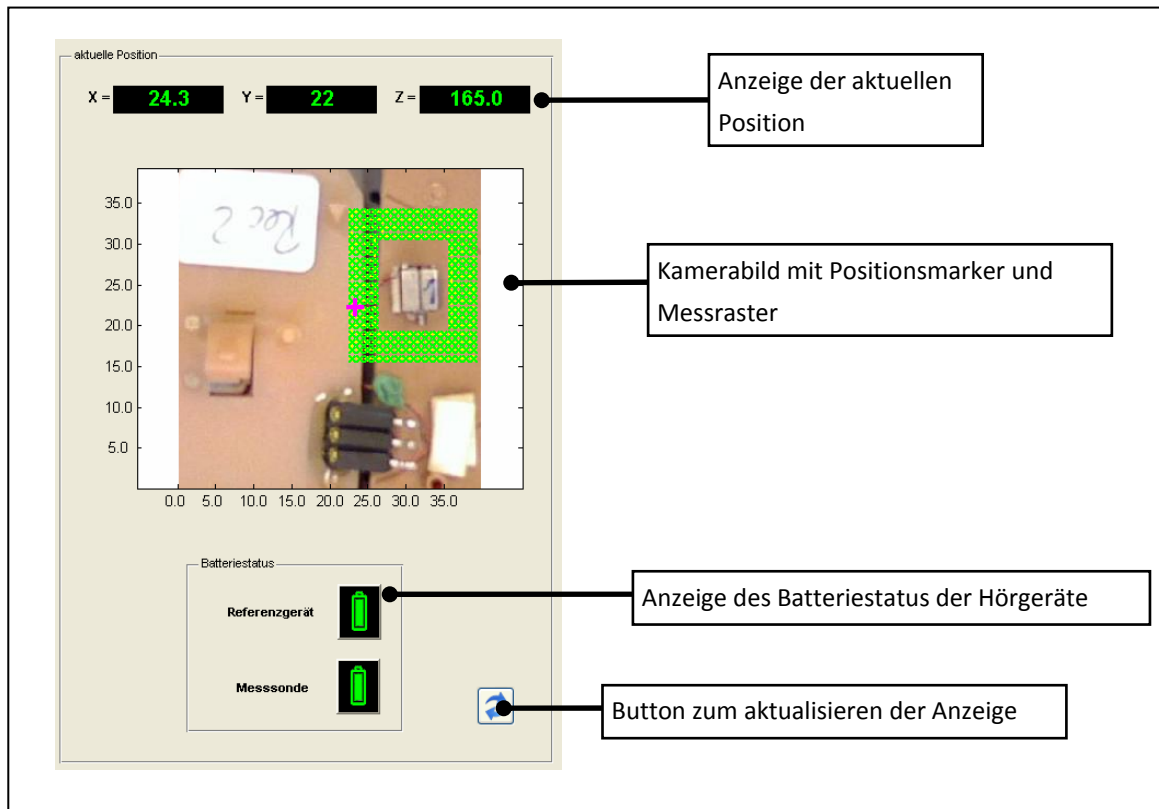


Abbildung 39: Anzeigebereich

(3) Freies Verfahren:

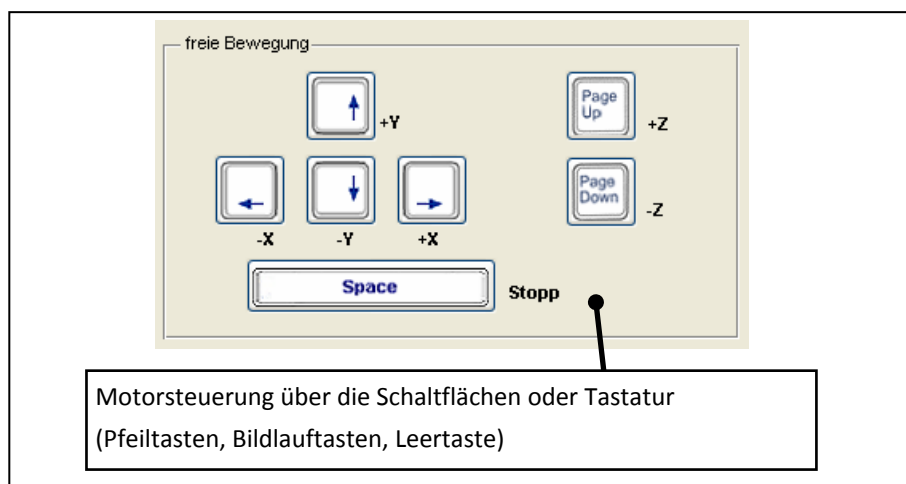


Abbildung 40: Feld „freie Bewegung“

(4) koordinatengesteuerte Bewegung:

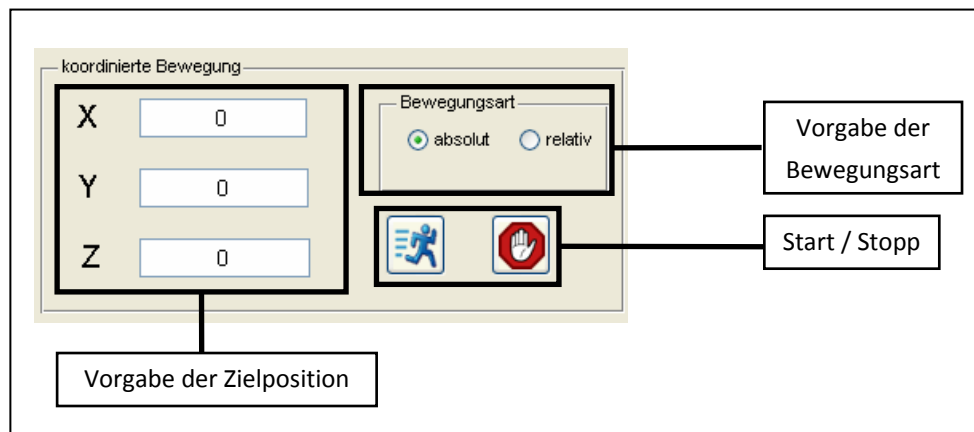


Abbildung 41: Feld „koordinierte Bewegung“

(5) Definition des Messbereiches:

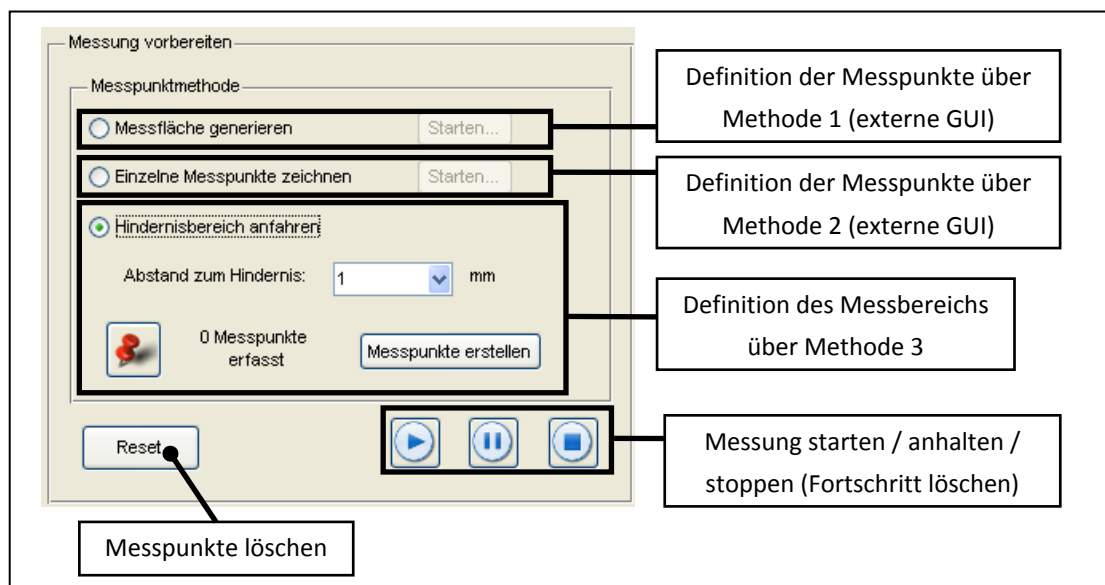


Abbildung 42: Feld „Messung vorbereiten“

II Einstellungsfenster

Das Fenster zur Anlagenkonfiguration wird beim Klick auf die Schaltfläche der Toolbar bzw. den entsprechenden Menüeintrag angezeigt

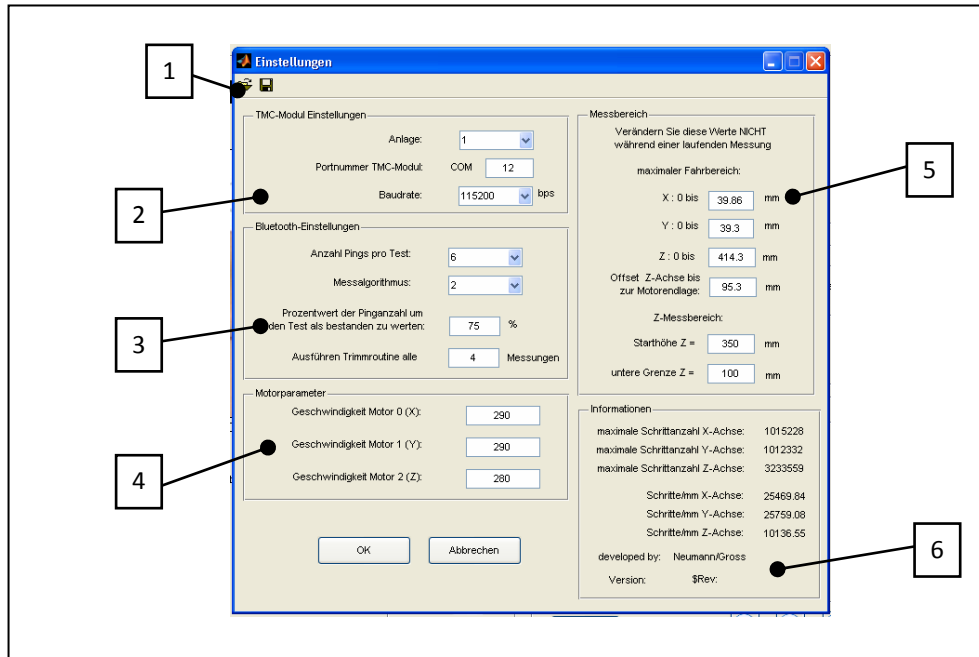


Abbildung 43: Einstellungsfenster

Tabelle 18: Komponenten des Einstellungsfensters

Nr.	Beschreibung
1	Laden und Speichern einer Konfigurationsdatei (*.ini)
2	Schnittstelleneinstellungen
3	Messeinstellungen
4	maximale Motorgeschwindigkeit
5	Messbereichsdimensionen
6	allgemeine Informationen

III Startfenster

Da der Benutzer die COM-Portnummer der Steuerkarte selbstständig ermitteln muss, wird zu diesem Zweck beim Start der Software ein Startfenster angezeigt. Des Weiteren kann der Nutzer in diesem Fenster den COM-Port und die Hörgeräte vorab testen.

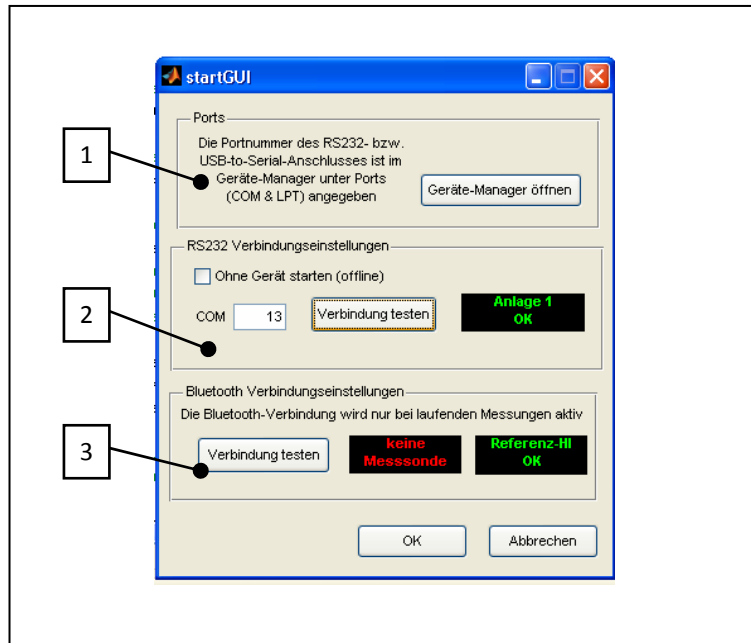


Abbildung 44: Startfenster

Tabelle 19: Komponenten des Startfensters

Nr.	Beschreibung
1	Öffnen des Windows Geräte Managers zur Ermittlung des COM-Ports
2	Einstellung der Portnummer und Test der Verbindung (mit Statusanzeige)
3	Test der Hörgeräte (mit Statusanzeige)

IV Auswertesoftware

Parallel zur Entwicklung der Anlagensoftware wurde abteilungsintern ein Auswertetool programmiert, mit dem die gespeicherten Messdaten visualisiert werden können.

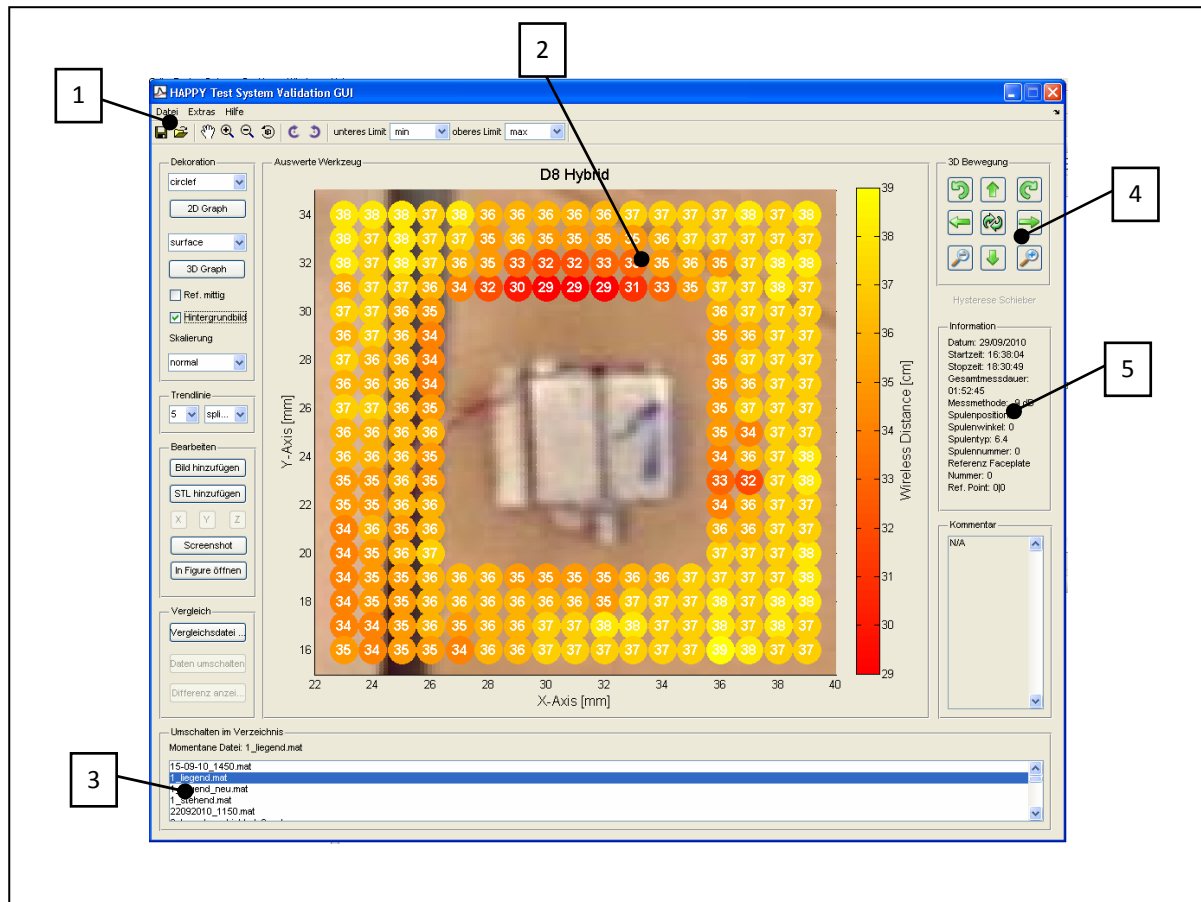


Abbildung 45: Auswertesoftware

Tabelle 20: Komponenten des Auswertetools

Nr.	Beschreibung
1	Toolbar und Menüleiste mit Standardfunktionen sowie Skalierungsschaltflächen
2	Messbereich mit den Z-Messwerten und dem Kamerabild des Messbereiches
3	Liste mit, im Ordner enthaltenen Messwertdateien
4	Zoom- und Drehfunktionen
5	Dateiinformatoren aus dem Header

7 Umsetzungen und Anforderungen im Vergleich

7.1 Hardware

7.1.1 Konfiguration

Forderung: Ein speicherbarer, fester Satz an Grundeinstellungen.

Umgesetzt wurde dies durch die Erstellung der Konfigurationsdateien (*.ini), die mit der Initialisierung geladen werden und im Systemordner gespeichert sind.

7.1.2 Feineinstellung der Messspule

Forderung: Eine stufenlose Höheneinstellung.

Umgesetzt wurde dies durch die neu angefertigte Trägervorrichtung.

7.1.3 Z-Achse

Forderung: Eine höhere Positioniergeschwindigkeit.

Umgesetzt wurde diese Forderung durch den stärkeren Z-Schrittmotor.

7.2 Messungen

7.2.1 Definition der Messpunkte

Forderung: Eine einfache Definition des Messbereichs.

Umsetzung: Der Benutzer kann nun über drei verschiedene Möglichkeiten den Messbereich definieren.

7.2.1.1 Pingpong-Test

Forderung: Eine automatische Durchführung und Auswertung des Tests.

Umsetzung: Der Pingpong-Test wird mittels des WLP nun automatisch durchgeführt und ausgewertet.

7.2.2 Trimmen der Hörgeräte

Forderung war ein automatischer Trimmvorgang während der Messung.

Umgesetzt wurde dies durch die automatische Trimmung in einem einstellbaren Intervall.

7.2.3 Messablauf

Forderung: Ein automatischer Messablauf.

Umsetzung: Nach der Festlegung des Messbereichs führt die Anlage die Messung selbständig durch und kann durch eine Statusanzeige überwacht werden.

7.2.4 Datenspeicherung und –auswertung

Forderung: Das automatische Speichern der Daten.

Umsetzung: Während der Messung werden an jedem Messpunkt die Daten zwischengespeichert. Über die Standardfunktionen der GUI kann der Nutzer nach Abschluss der Messung die Daten in verschiedenen Formaten abspeichern. Die Auswertung in Visualisierung kann mit dem Auswertetool vorgenommen werden.

7.3 PC-Benutzeroberfläche

7.3.1 Bedienung

Forderung war eine übersichtliche und einfach zu bedienende Oberfläche.

Umsetzung: Die neue Benutzeroberfläche wurde mit allen wichtigen Funktionen (auch in Absprache mit den späteren Benutzern) gestaltet.

7.3.2 Fahrfunktionen

Forderung war die unkomplizierte Positionierung der Achsen.

Umgesetzt wurde dies durch die Freifahr- und Positionierfunktionen in der GUI.

7.3.3 Statusüberwachung

Forderung: Die Anzeige der aktuellen Position sowie die Überwachung des Batteriestatus der Hörgeräte.

Umsetzung: Die GUI enthält einen Positionsmarker sowie numerische Positionsanzeigen. Die Batteriekapazität wird während der Messung aktualisiert und kann durch Schaltflächen abgerufen werden.

7.3.4 Einstellmöglichkeiten

Forderung war die einfache Veränderung der wichtigsten Einstellungen.

Umsetzung: Alle relevanten Einstellungen können über ein entsprechendes Fenster verändert werden.

8 Ausblick und Erweiterungen

Im Hinblick auf Verbesserung und Steigerung der Effizienz bietet die Anlage noch eine Vielzahl an Erweiterungsmöglichkeiten. Während der Test- und Betriebsphase wurden daher folgende Erweiterungspunkte aufgestellt:

Änderung der Z-Spindel

Durch den äußerst niedrigen Wirkungsgrad der Z-Achse (Vgl. 3.12) bietet es sich hier an, eine Trapezgewindespindel oder einen Kugelumlauftrieb zu verwenden. Ein Konzept dieser Art bewirke zusätzlich zur Steigerung der Verfahrgeschwindigkeit eine Verringerung des Verschleißes. Nachteil wäre hierbei das weitere Anfallen von Fertigungs- und Materialkosten.

Neigung der Z-Achse

Um zusätzlich zum Einfluss des axialen Abstandes den Einfluss des Winkels der beiden Spulen zueinander untersuchen zu können, kann die Neigung Z-Achse (ggf. anhand eines weiteren Schrittmotors) verstellt werden.

Ferndiagnose

Unter Verwendung eines vorher eingerichteten Webserver kann der aktuelle Messfortschritt online gestellt werden, um die Überwachung ohne direkte Verbindung zur Anlage zu ermöglichen. Des Weiteren kann die Email-Sende-Funktion von MATLAB verwendet werden, um z.B. Fehler zu melden. Bisher scheiterte dies jedoch an den Sicherheits- und Zugriffsrichtlinien der Webservern.

Referenzmessung

Um äußere Störungen, z.B. durch benachbarte Verkabelungen, detektieren zu können, kann neben dem Messaufbau eine regelmäßige Referenzmessung mit zwei unabhängigen Hörgeräten durchgeführt werden. Dafür sollen zwei optimal funktionierende Geräte (ohne interne Störwirkung) im Maximalabstand zueinander entsprechend dem Pingpong-Test abgefragt werden.

Kamera

Die Bildqualität der aktuell verwendeten Kamera ist für die Größe des Testfaceplates nicht optimal. Für bessere Fokussierung und Auflösung wäre daher eine Kamera mit Autofokusfunktion und Beleuchtung vorteilhaft.

Bildskalierung

Die manuelle Skalierung des Kamerabildes ist aufgrund der eventuellen Ungenauigkeit häufige Fehlerquelle. Um dies zu unterdrücken, könnte an einer automatischen Skalierung (z.B. durch automatische Erkennung der Markierung) gearbeitet werden.

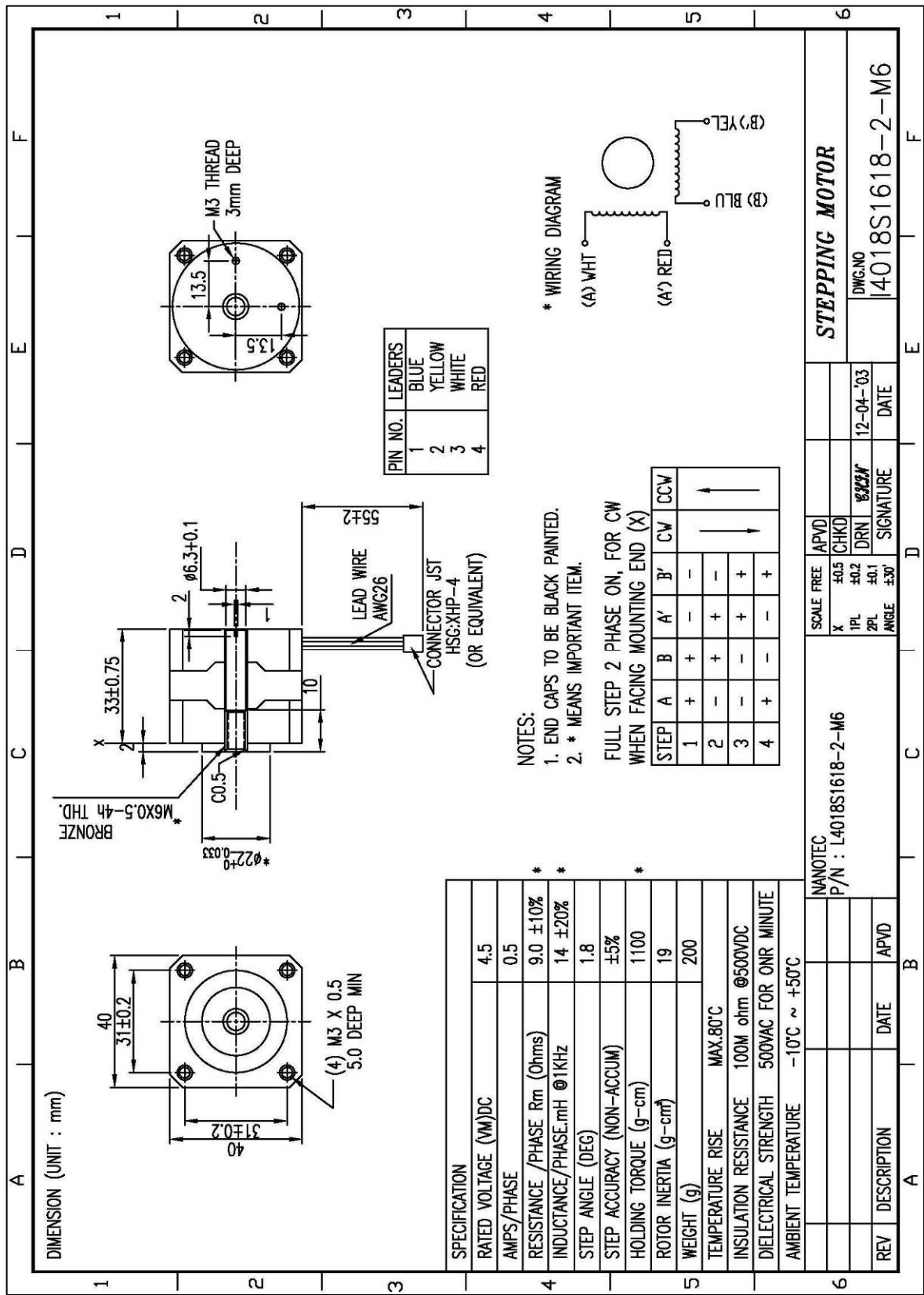
Erweiterung der Algorithmen

Um die Flexibilität und Aussagefähigkeit weiter zu erhöhen, können noch weitere Pingpong-Testalgorithmen erstellt werden. Beispielsweise kann eine Hysteresemessung erstellt werden, bei der die Pingpong-Testergebnisse jedes Z-Punktes gespeichert werden um den Bereich zwischen 0 und 100 % zu erfassen.

Anhang

A Datenblätter

A.1 Nanotec L4018S161



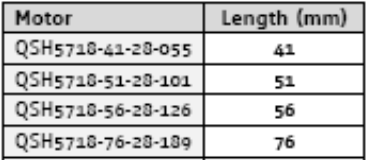
A.2 Trinamic QSH5718-51-28-101 (Auszug aus [15])

Characteristics:

- NEMA 23 mounting configuration
- 6.35mm axis diameter, 20mm axis length
- step angle 1.8
- optimized for microstep operation
- optimized fit for TMC239/TMC249/TMC262 based driver circuits
- 4 wire connection
- CE approved
- flange max. 56.5mm x 56.5mm
- D-cut of 15mm length and 0.5mm depth
- up to 75V recommended operation voltage

Specifications	Parameter	Units	QSH5718			
			-41-28-055	-51-28-101	56-28-126	-76-28-189
Number of Leads		N°	4	4	4	4
Step Angle		°	1.8	1.8	1.8	1.8
Step Angle Accuracy		%	5	5	5	5
Rated Voltage	V_{RATED}	V	2	2.3	2.5	3.2
Rated Phase Current	$I_{\text{RMS RATED}}$	A	2.8	2.8	2.8	2.8
Phase Resistance at 20°C	R_{COIL}	Ω	0.7	0.83	0.9	1.13
Phase Inductance (typ.)		mH	1.4	2.2	2.5	3.6
Holding Torque		Nm	0.55	1.01	1.26	1.89
Detent Torque		Nm	0.020	0.035	0.039	0.066
Rotor Inertia		g cm ²	120	275	300	480
Insulation Class			B	B	B	B
Max. applicable voltage		V	75	75	75	75
Max. radial force (20mm from front flange)		N	75	75	75	75
Max. axial force		N	15	15	15	15
Weight		kg	0.45	0.65	0.7	1
Length		mm	41	51	56	76
Temp. Rise (rated current, 2 phase on)		°C	+80 max	+80 max	+80 max	+80 max
Ambient Temperature		°C	-20 +50	-20 +50	-20 +50	-20 +50

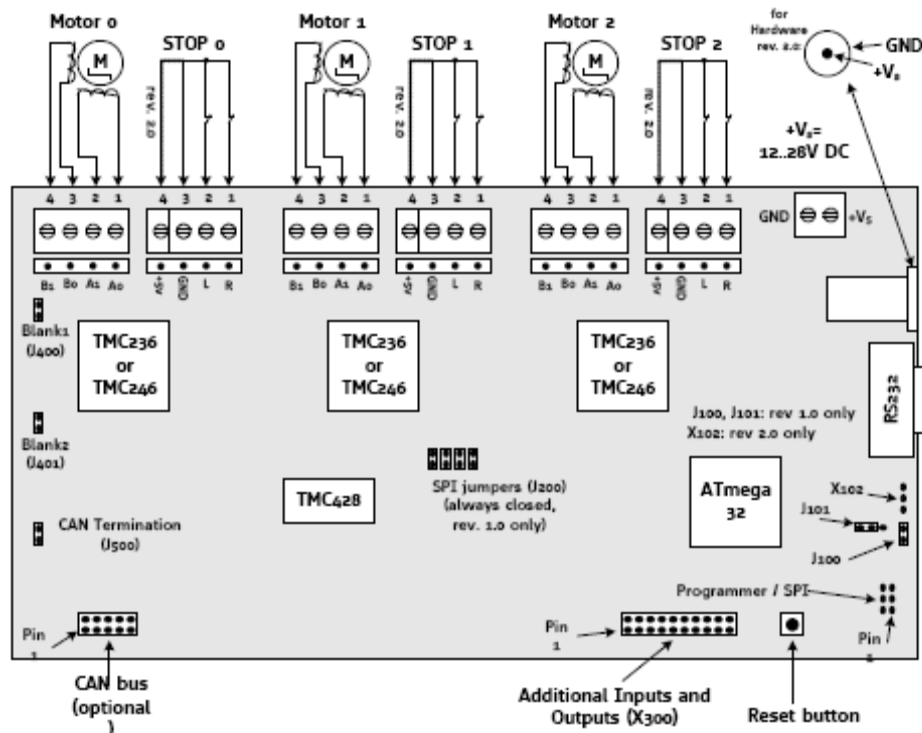
Table 2.1: Specifications of QSH5718-41-28-055, QSH5718-51-28-101, QSH5718-56-28-126, and QSH5718-76-28-189



A.3 Trinamic TMCM-310 (Auszug aus [4])

Symbol	Parameter	Min	Typ	Max	Unit
$V_S (V_{1.x})$	DC Power supply voltage for operation	9	12... 28	28	V
$V_S (V_{2.0})$	DC Power supply voltage for operation	8	12... 28	34	V
I_{COIL}	Motor coil current for sine wave peak (chopper regulated, adjustable via software)	0	0.3... 1.5	1.5	A
f_{CHOP}	Motor chopper frequency		25		kHz
I_S	Power supply current (per motor)		$\ll I_{COIL}$	$1.4 \cdot I_{COIL}$	A
U_{+5V}	+5V output (max. 150mA load)	4.8	5.0	5.2	V
V_{INPROT}	Input voltage for StopL, StopR, GPIO (internal protection diodes)	-0.5	0... 5	$V_{+5V} + 0.5$	V
V_{ANA}	INx analog measurement range		0... 5		V
V_{INLO}	INx, StopL, StopR low level input		0	0.9	V
V_{INHI}	INx, StopL, StopR high level input (integrated 10k pullup to +5V for Stop)	2	5		V
I_{OUTx}	OUTx max +/- output current (CMOS output) (sum for all outputs max. 50mA)			+/- 20	mA
T_{ENV}	Environment temperature at rated current (no cooling)	-40		+80	°C

Table 5.1: Operational Ratings



B TMCL-Befehlsliste (Auszug aus [5])

Mnemonic	Command number	Meaning
ROL	2	Rotate left
ROR	1	Rotate right
MVP	4	Move to position
MST	3	Motor stop
RFS	13	Reference search
SCO	30	Store coordinate
CCO	32	Capture coordinate
GCO	31	Get coordinate

Mnemonic	Command number	Meaning
SAP	5	Set axis parameter
GAP	6	Get axis parameter
STAP	7	Store axis parameter into EEPROM
RSAP	8	Restore axis parameter from EEPROM
SGP	9	Set global parameter
GGP	10	Get global parameter
STGP	11	Store global parameter into EEPROM
RSGP	12	Restore global parameter from EEPROM

Mnemonic	Command number	Meaning
SIO	14	Set output
GIO	15	Get input
SAC	29	Access to external SPI device

Mnemonic	Command number	Meaning
JA	22	Jump always
JC	21	Jump conditional
COMP	20	Compare accumulator with constant value
CLE	36	Clear error flags
CSUB	23	Call subroutine
RSUB	24	Return from subroutine
WAIT	27	Wait for a specified event
STOP	28	End of a TMCL program

Mnemonic	Command number	Meaning
CALC	19	Calculate using the accumulator and a constant value
CALCX	33	Calculate using the accumulator and the X register
AAP	34	Copy accumulator to an axis parameter
AGP	35	Copy accumulator to a global parameter

C MATLAB-Funktionsliste

Name	Beschreibung	Autor ¹
addImage	Hinzufügen eines Bildes zur Messdatenstruktur	Neumann/Groß
addSTL	STL-Datei zum Bild hinzufügen	Neumann
adjustImage	Bild skalieren und drehen	Neumann/Groß
astar_mod	Modifizierter A*-Algorithmus zur Berechnung eines Pfades aus einer Koordinatenmatrix, UF für findWaypoints	[20]
automatedMode	Hauptfunktion des Messablaufs	Groß
base64img	Codieren einer MATLAB-Bildmatrix in mime-Format für Emails	[20]
base64imgdecode	Decodieren von MIME-Daten in eine MATLAB-Bildmatrix	[20]
bt_init	Initialisierung der WLP-Verbindung zum HI	Groß
bt_measure	Mess- und Positionierfunktion	Groß
bytes2val	Umrechnung Bytefolge (von TMCM) in Zahlenwert	Groß
calibrateCam	Kameradrehung korrigieren	Neumann/Groß
calibrateCamGUI	Hauptfunktion der GUI zur Korrektur der Kameradrehung	Neumann/Groß
change_value	Variablenwert in Textdatei ändern (UF für pdf-Datenerstellung)	[20]
compareMeas	Messdaten vergleichen	Neumann
configGUI	Hauptfunktion des Einstellungsfensters	Groß
convert	Umrechnungsfunktion für Pixel/mm/cm/Schritte	Groß
convertOldData	Alte Messdaten für die Auswertung konvertieren	Neumann/Groß

¹ Name des Softwareautors bzw. die Angabe der Quelle bei Verwendung vorgefertigter Funktionen aus dem Internet

copyFigures	UF zum Export von Figures in doc-Dateien	Neumann
correctPoints	Ausreißer aus den Messdaten entfernen	Neumann
create_word_figures	Export von Figures in doc-Format	Neumann
createHTML	HTML-Datei aus Messdaten erstellen	Neumann/Groß
dataValidator	Korrekturfkt. für defekte Messdaten	Neumann
disp_path	Plotten der Messpunkte in Anzeige	Groß
distance	UF für A*-Algorithmus	[20]
drawCalib	Bild zuschneiden	Neumann/Groß
drawFrame	Messbereich zeichnen (Methode 1)	Neumann/Groß
drawObstacles	Hindernisbereich zeichnen (Methode 3)	Neumann/Groß
drawSVG	Erzeugen einer SVG-Grafik aus Messdaten (z.B. zur Verwendung in MS Internet Explorer)	Neumann/Groß
eps2pdf	EPS-Datei für pdf-Dateien umwandeln	[20]
expand_array	UF für A*-Algorithmus	[20]
export_fig	Messdiagramm in Bitmap-Bild exportieren	[20]
exportData	Messdaten in versch. Dateiformate exportieren	Neumann/Groß
findjobj	Nebenfunktion zur Darstellung Java-basierter Elemente in GUI	[18]
findWaypoints	Messpunkte um Hindernisbereich erzeugen (Methode 3)	Neumann/Groß
fix_lines	Liniendarstellung für EPS-Dateien verbessern	[20]
generateData	Leere Messdatenstruktur erzeugen	Neumann/Groß
get_batt	Batteriestatus der Hörgeräte abfragen	Groß
get_Pos	Aktuelle Position abfragen	Groß
ghostscript	UF zum Export von Figures in versch. Grafikformate	[20]
go2coord	Koordinierte Bewegung der X- und Y-Achse	Neumann/Groß
go2z	Koordinierte Bewegung der Z-Achse	Neumann/Groß
headerGUI	Hauptfunktion der GUI zur Header-Bearbeitung	Groß

initial	Initialisierungsfunktion	Groß
insert_open	UF für A*-Algorithmus	[20]
limit_status	Abfrage der Endschalter	Groß
loadLanguage	Sprachdatei für Auswertesoftware laden	Neumann
mainGUI	Hauptfunktion des Hauptfensters	Groß
min_fn	UF für A*-Algorithmus	[20]
node_index	UF für A*-Algorithmus	[20]
obst2gemuese	Erzeugen eines Rechtecks um min. 3 Punkte (für Methode 3)	Groß
pathfinderGUI	Hauptfunktion der GUI von Methode 2	Neumann(Groß
pdf2eps	UF zum Export von Figures in versch. Grafikformate	[20]
pdftops	UF zum Export von Figures in versch. Grafikformate	[20]
print2array	UF zum Export von Figures in versch. Grafikformate	[20]
print2eps	UF zum Export von Figures in versch. Grafikformate	[20]
readConfig	Konfigurationsdatei (*.ini) einlesen	Groß
readHeaderValues	Headerdaten einlesen (für Bearbeitung)	Neumann/Groß
recentFiles	Erzeugen einer Liste der zuletzt geöffneten Dateien (für Auswertetool)	Neumann
recordPath	Einzelne Messpunkte festlegen (Methode 2)	Groß
ref_fahrt	Referenzfahrt durchführen	Groß
refresh_disp	Anzeige aktualisieren	Groß
rotateMatrix	Messpfadmatrix drehen	Neumann
send	Kommunikationsfkt. für TMCM	Groß
serial_reset	Fehlerfunktion für COM-Schnittstelle	Groß
setBackground	Hintergrundbild hinzufügen (Auswertetool)	Neumann

setCustomScaling	Einheitliche Skalierung für Hintergrundbilder (Auswertetool)	Neumann
setTrendline	Trendlinie erzeugen (Auswertetool)	Neumann
startGUI	Hauptfunktion des Startfensters	Groß
startup	MATLAB Startfunktion	Neumann/Groß
statusbar	Fkt. zum Hinzufügen einer Statusleiste in die GUI	[18]
STL_Import	Import von STL-Dateien	[20]
systray	Ändern des Systemicons im Fenster	[18]
takeImage	Webcambild aufnehmen	Neumann/Groß
trim	Trimmen der Hörgeräte	Groß
turnSTL	Import einer STL-Bibliothek	Neumann
val2bytes	Umrechnung eines Zahlenwertes in Bytefolge (für TMCL-Bytefolge)	Groß
validation2D	Ansichtsmodus für Auswertetool	Neumann
validation3D	Ansichtsmodus für Auswertetool	Neumann
validationGUI	Hauptfunktion des Auswertetools	Neumann
validationHyst	Auswertefunktion für Hysteresemessung	Neumann
write2log	Logfileeintrag schreiben	Neumann/Groß
writeConfig	Konfigurationsdatei erzeugen	Groß
xmlWaypointsExport	Export der Messpunkte in xml-Datei	Neumann
xmlWaypointsImport	Import von Messpunkten aus xml-Datei	Neumann

D Datenträger

Inhalt:

- Software
- Installationsanleitung mit Beschreibung externer Dateien
- Quellen (sofern digitalisiert verfügbar)
- HTML-Dokumentation aller Funktionen
- Diplomarbeit als pdf-Dokument

Abkürzungsverzeichnis

EPS	Grafikformat für Internet-Nachrichten (engl.: Enabled PostScript)
FEKO	Computerprogramm zur Berechnung elektromagnetischer Felder
GUI	Grafische Benutzeroberfläche (engl.: Graphical User Interface)
HdO	Hinter-dem-Ohr-Hörgerät
HI	Hörgerät (engl.: Hearing Instrument)
IdO	In-dem-Ohr-Hörgerät
MS	Messsonde
RG	Referenzgerät
SAT	Siemens Audiologische Technik GmbH
STL	Programmbibliothek für C++ (engl.: Standard Template Library)
SVG	Vektorgrafik auf XML-Basis (engl.: Scalable Vector Graphics)
SWT	Standalone Wireless Tester
TMCL-IDE	Trinamic Programmiersprache (Trinamic Motion Control – Integrated Development Environment)
TMCM	Trinamic Schrittmotorsteuerung (Trinamic Motion Control Module)
UF	Unterfunktion
WLP	Wireless Programmer
XML	Extensible Markup Language

Literaturverzeichnis

- [1] Siemens Audiologische Technik GmbH: *Siemens Technikbroschüre für Hörgeräteakustiker*, 04/2010
- [2] Dr. Volker Gebhardt: *D6 Wireless Implementation Guidelines*, Erlangen, 05/2009
- [3] Gottfried Rückerl, *WL1-Präsentation* (MS Powerpoint), Erlangen, 05/2006
- [4] Trinamic Motion Control GmbH: *TMCM-310 Hardware Manual*, 09/2009
- [5] Trinamic Motion Control GmbH: *TMCL Reference and Programming Manual*, 05/2008
- [6] Voltcraft: *Bedienungsanleitung Voltcraft PA-300*
http://www.produktinfo.conrad.com/datenblaetter/500000-524999/518302-an-01-ml-STECKERNETZGER_PA300_de_en_fr_nl.pdf,
11/2008
- [7] Nanotec Electronic GmbH & Co. KG: Datenblatt L4018S1618-2-M6, 04/2003
- [8] Nanotec Electronic GmbH & Co. KG: Katalog Linearaktuatoren, 01/2009
- [9] Voltcraft: *Bedienungsanleitung Voltcraft SNG-24*, 5/2010
- [10] NEFF Gewindetriebe GmbH: *Katalog Gewindetriebe, Spindelhubgetriebe*,
<http://www.neff-gewindetriebe.de/PDF/Katalog%20Neff%20Gewindetriebe%202009.pdf>
10/2008
- [11] Keller und Kalmbach: *Technisches Handbuch, Schrauben, Formen und Ausführungen*,
<http://www.keller-kalmbach.de/fileadmin/images/web/pdf-dokumente/1-1-Schrauben-Formen-und-Ausfuehrungen.pdf>
07/2006

- [12] <http://www.tedata.com/2211.0.html>,
Zugriff: 10/2010

- [13] ACRYLGLAS-Technik Werner Winkler GmbH & Co KG: *Datenblatt PVC*,
http://www.acrylglastechnik.at/Datenblatt/Db_PVC.pdf,
06/2006

- [14] <http://www.iso-gewinde.at/>,
Zugriff: 10/2010

- [15] Trinamic Motion Control GmbH: *QMot QSH5718 family Manual*, 10/2010

- [16] Mathworks Inc: *MATLAB 7.5.0 R2007b HTML-Help*, 2007

- [17] Gunter Sauer, Siemens Audiologische Technik GmbH: *Target Specification DIANA 6*,
Erlangen, 11/2008

- [18] <http://undocumentedmatlab.com/>,
Zugriff: 06/2010

- [19] Dieter Meschede: *Gerthsen Physik*, Springer, Berlin, 04/2005

- [20] <http://www.mathworks.com/matlabcentral/fileexchange>,
Zugriff: 09/2010

Erklärung

Die vorliegende Abschlussarbeit wurde von mir selbständig, ohne fremde Hilfe und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt. Wörtliche und sinngemäße Zitate im Text sind als solche gekennzeichnet.

Erlangen, November 2010

Matthias Groß

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die zum Gelingen dieser Diplomarbeit beigetragen haben.

Herrn Prof. Dr.-Ing. Dietmar Römer danke ich für die hochschulseitige Betreuung.

Ein sehr großer Dank gilt meinem Betreuer in der Siemens Audiologischen Technik GmbH, Herrn Dr. Volker Gebhardt. Ich danke ihm für die fachliche Unterstützung und die stets engagierte Betreuung.

Herzlich danken möchte ich auch dem Team der SAT-Werkstatt für die Investition zahlreicher Stunden in den Bau der beiden Anlagen.

Zudem möchte ich mich bei allen Mitarbeitern der Abteilung HW3 bedanken, die mich durch hilfreiche Tipps und aufheiternde Momente jederzeit unterstützten.

Großer Dank gilt auch meiner Familie, besonders meiner Nichte Lena, für die ausgleichenden Momente an den Wochenenden.

Tausend Dank, Christian!

Der größte Dank gilt zuletzt Herrn Benjamin Neumann, ohne dessen aufheiternde Tipps und fachliches Know-how diese Arbeit in dieser Form nicht entstanden wäre.